Tools for Collaborative Software Inspection Management

Lai Poh Fatt
WEK 020090

Under the Supervision of
Mrs.Norazlina Khamis

Moderator
Ass. Prof. Dr. Lee Sai Peck

This project is submitted to the
Faculty of Computer Science and Information Technology
University of Malaya

In partial fulfilment of the requirement for the degree of
Bachelor of Computer Science

Session 2004/2005

Abstract

InspectionManager is a web based system that is used by the Inspection Moderator, Inspector and Document Author. Inspection Moderator can manage his inspection process properly here. Inspection can set the inspection objective, invite the inspectors, review the defects discovery logs, schedule for the inspection overview or inspection meeting and etc. Meanwhile the inspector can retrieve the to be inspected document here, have their preparation for meeting by finding potential defects in the document and log the defects through InspectionManager. The Author can upload their document here and correct, explain the defects. The system is also available online 24 hours a day throughout the year.

The aim of this project is to build a software inspection tool to replace the existing manual process in order to upgrade the business strategies. The core objective of this system is to build a web based application which allowing all tasks to be done through electronic. Researches are done on the existing system to get a better view of traditional software inspection process. The Rational Unified Process is used to develop the InspectionManager. InspectionManager is an active web page solution that needs special development tools and languages such as ASP.NET, Internet Information Server (IIS), Macromedia Dreamweaver MX and others.

InspectionManager is developed using Object-Oriented Analysis and Design which provides a particularly fine basis for internal or structural design.

Acknowledgement

The development of this online software inspection tool project was carried along with advices, assistance, contributions and ideas from many individual.

First and foremost, I would like to express my utmost gratitude to my supervisor, Mrs. Noazlina Khamis for the guidance throughout the development of my project. Not forgetting, my special thanks to Dr. Lee Sai Peck, my project moderator for spending precious time to moderate this project

Special thanks also go to my fellow course mates, friends and senior for sharing their time and knowledge with me. Their supports and motivations are deeply appreciated. I feel grateful to have their encouragements whenever I encounter difficulty.

Last but not least, I would like to express my heartiest appreciation to my beloved parents and siblings for all their love, moral supports and encouragement throughout this period. Their kindness will always get a special place in the bottom of my heart.

Table of Content

List of Figure

List of Table

Chapter 1- Introduction

1.1 Project Overview

Software inspection is a detailed review of a small amount of material by technically competent peers with the goal of detecting faults (M.Fagan, 1976). It is proven as a cost-effective way in produce a quality software product. Software inspection process will assembles all the people work together in an iterative way at the same time and at the same place. But increasing trend in large-scale software development will bring to a stage that software inspection solutions have varying a trade-off in tome, cost and effectiveness (Perpich, 1997). This brings to a stage that a need for collaborative software inspection management system to make sure that the software inspection process is efficient and effective, cost saving and easy to maintain.

InspectionManager is a web-based tool for collaborative software inspection management, which caters for software house developers, software inspection consultancies, researcher and any other organizations, which need software inspections. It provides comprehensive features and standard templates for the software inspection team in manage the inspection process. It is to assist software inspection Moderator in planning, over viewing, preparing, carry out inspection meeting, collecting software artefacts and repairing through out the software inspection process in a systematic and cost-effective way. The system's modules have been designed based on good and essential features identified from existing systems available on the Web. InspectionManager hopes to meet the needs of software inspector/moderator in any software organization and benefit from all the high-end features of this system.

1.2 Problem Statement

It is recognized that the inspection over software artifacts such as requirement specification, design and codes as a means to detecting and removing defects from software. Through out the process, a software inspection team is form and generally leads by Moderator. Over the past 20 years, the software inspection process was done manually. There are several problems that face by the organization during software inspections:

Cost

Cost is needed throughout the software inspection process and it shouldn't be more than the cost needed to correct the software product without gone through this process. The costs involved the cost for train and Select Inspection Team: Moderator, reviewer, recorder and other team members. Besides that, costs also include the payment for the Software Inspection Team member's effort and preparation for inspection meeting.

Documentation and others

Cost also needed for efforts in entering the forms, preparing the reports, data entry in the database and efforts in analyze the data.

Geographically separated team members

Popular trend in large-scale software development which use of geographically separated groups in development leads to a problem during software inspection. As

software inspection process needs authors and reviewer/inspection attending meetings, geographically separated member will suffer in scheduling for meeting and activities.

Time consuming and inefficient

The existing process always involved a lot of paperwork as most of the records are done manually. It is time consuming to track the specific record that is needed. They can also cause many problems, for example difficult to read the hand writing, human mistake and others.

Due to these problems, the whole software inspection processes have trade-off in cost, time and effectiveness. Because of the problems stated above, the aim of this project is to develop a computerized system that can solve these problems.

1.3 Project Objectives

InspectionManager is developed to benefit all personnel involved in collaborative software inspection. The system is expected to provide a highly effective management solution for software inspection team members.

The objectives hoped to be achieved by InspectionManager are:

1. To provide Inspection Moderator a tool to select and record his inspection team.
2. To provide electronic way to record defects identified for software inspector during preparation phase.
3. To gather all records from every inspectors and perform analysis to find out the most possible potential defects to save time and effort during inspection meeting.
4. To provide software inspection team easy accesses to software inspection artifacts and information for further inspection process
5. To implement Information Technology as an electronic solutions to establish software inspection process compare with traditional paper-based software inspection.

1.4 Project Scope

The most important goal of the development of InspectionManager is to provide a tool to support collaborative software inspection management. The system aims to manage the process of software inspection: Planning, Overview, Preparation, Inspection Meeting, Rework and Follow-up in an efficient and effective way. Since there are a plenty of material to be inspected, this system will only cover inspection on **requirements document.** The system will only support English as a single communication language. All the information provided will be in English.

Target user:

1. Inspection Moderator
2. Inspection Inspector
3. Requirements document's Author

The system is expected to be developed in six modules:

Planning Module

Planning purpose is for organize and plan resources for inspection. The module will allowed the Moderator to verify the product need to be inspected, select the inspection team members, and schedule the date for an overview meeting.

Overview Module

Overview purpose is to educate and inform the inspector about the work product. This module will show all the information about the overview meeting which are: The work product to be overviewed, list of inspector as well as the time and venue.

Preparation Module

Each inspector works individually to inspect the work products and identified the potential defects. All the defects will add into the database through this module.

Inspection Meeting Module

All the defects identified by inspector will gather and discuss here. The duplicate defects are list in the most possible potential defects and will consider reworking. The defects will be stored in database.

Rework Module

Author will correct the defect identified and response update the defect list here.

Follow-up Module

Moderator verified the work product here and completes the Inspection Summary Report.

1.5 Expected Outcome

The expected outcomes of the project are stated below:

- The system is available on-line.
- The system is able to ensure only authorized user can access the webpage and use the function.
- The system can perform the function such as manage the log inspection defects and access database.
- Provide a more effective and efficient way to manage software inspection process.
- The system is able to expand to allow future enhancement and additional functions.

1.6 Project Schedule

To implement time management and systematic working throughout the system development time frame, a project schedule is carefully planned. This is to ensure achievement of the outlined project objectives at the end of the project development period. At the first phase of the development, system study and literature review are carried out to collect information about the system for the planning of a good development strategy. A project proposal is prepared and a viva session is held. The system is ready to design after the system requirement analysis has been done. Then comes to the second stage where system is developed. The project chedule for InspectionManager is as shown in figure 1.1:

ID	Task Name	Start	Finish	Duration
1	Project Definition & Introduction	7/2/2004	7/22/2004	15d
2	Research & Literature Review	7/19/2004	8/5/2004	14d
3	Methodology	8/2/2004	8/16/2004	11d
4	System Analysis	8/23/2004	9/10/2004	15d
5	System Design	9/8/2004	10/15/2004	28d
6	System Implemetation	9/10/2004	1/12/2005	89d
7	System Testing	9/13/2004	2/11/2005	110d
8	System Evaluation	1/20/2005	2/18/2005	22d
9	Supervisor Consultation	7/2/2004	2/24/2005	170d
10	Documentation	7/9/2004	2/24/2005	165d

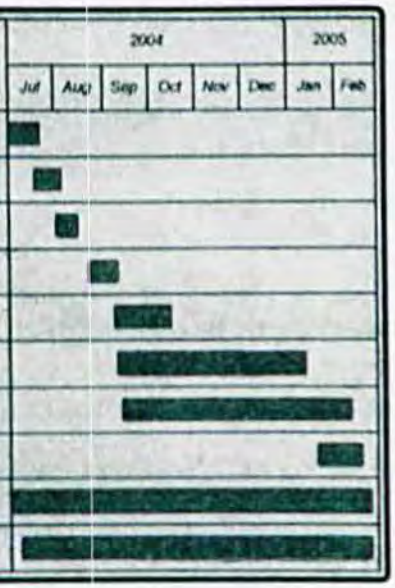


Figure 4-1: Project Schedule

1.7 Report Layout

Chapter 1: Introduce

This chapter consists of introduction to the system, objective, project scopes, project schedule and overview of whole documentation.

Chapter 2: Literature Review

This chapter covers the definition of the system, and some explanation on topics researched and studied that are relevant to this project. This chapter also contains some review about the technology going to be used to develop the system.

Chapter 3: Methodology

This chapter covers the software development life cycle, methodology used to develop the system and the information gathering about this system.

Chapter 4: System Analysis

This chapter covers the analysis phase of the project. It includes system requirement, functional and non-functional requirement, and hardware and software requirement and describes the tools and technology to be used.

Chapter 5: System Design

This chapter consists of the introduction of system, system architecture, system functionality design, and database and interface design.

Chapter 2: Literature Review

A research and literature review to find out more information about the system is important. With that, the overview of the system will be seen. The material bellows are concluded following some efforts such as studying and looking at the similar projects from faculty's document room, looking for reference books from library, extra reference materials from Internet and purchasing some useful related books.

2.1 Domain Studies

A software inspection is a process of defect detection, defect elimination, and correction verification carried out by a small group during the development life cycle. A defect is any error, non-conformance, or failure to satisfy a requirement in a product. The goal of formal inspections is to ensure that defects are fixed early in the life cycle rather than late, when they are harder to find and more costly to fix. Formal inspections are held throughout the software development life cycle phases to correct the products and to provide a short feedback loop for authors.

2.1.1 Definition and Terminologies

Finding and fixing a software problem after delivery is often 100 times more expensive than finding and fixing it during the requirements and design phase (Schach, 2005). The defects should be fixed as early as possible to save cost and reduce development works. During a software development process, avoidable rework accounts for 40-50% of development (Boehm, 1987).

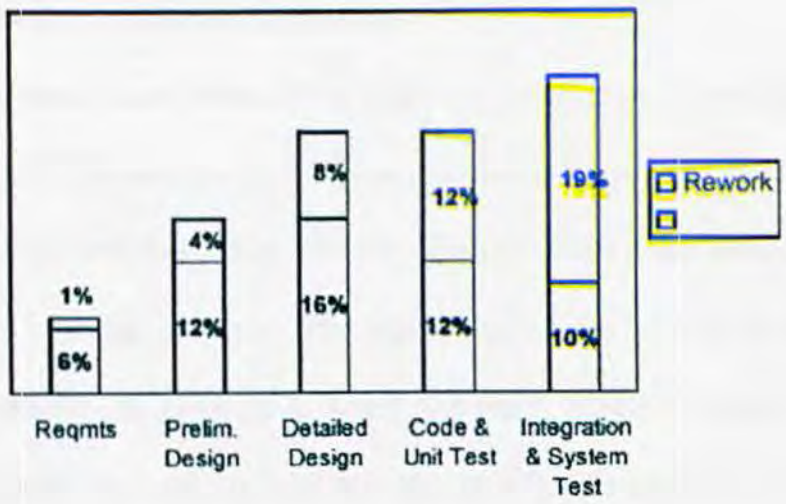


Figure 2.1 Rework accounts during software development.

Michael Fagan at IBM developed and introduced software inspections for the purpose to detect software defects and process improvement (Fagan, 1976). A defect is any error, non-conformances, or failure to satisfy a requirement in a product. Software inspections are different from other peer reviews because it is highly structured and below are software inspection's characteristics:

- A phased process to follow, which contains planning, overview, preparation, meeting, rework and follow up)
- Roles performed by peers during review, which are moderator, author, recorder, reader and inspector)
- A reading toolset to guide the review activity, for example, defect taxonomies, product checklist or scenario-based reading techniques
- Forms and report templates to collect product and process data.

2.1.1.1 The benefit of Software Inspections

By adopting software inspections during software developing process, especially some of the giant software product, early defect detection improves product quality and reduces avoidable rework down to 10-20% (Ebenua, 1994). By reduce the avoidable rework, a lot of cost can be saved. The cost saved can be an extra budget during the development process to produce a better software product. Besides that, usually software inspection uncovers defects that may not be discovered by testing. Cost is much more expensive if such a defect didn't detect during testing phases and have to be fixed after the software is released. Moreover, the software inspection serves for training. During the software inspection process, participants learn to avoid systematic defects and they also learn from high-quality products. Newcomers also learn about domain, product and standards. By the way, when noticed that the product will be reviewed by someone else, system developers are motivated to polish their works to a higher shine such as making style consistent, improving clarity. So, reviewing software product improves quality by encouraging system developers to work harder.

2.1.1.2 The Inspection Team

Before the software inspection task is carried out, an inspection team has to be formed. The inspection team consists of a group of individuals that work together to study and analyse each work product of a development activity in order to detect and remove defects. Five different procedural roles are assigned to the individuals that make up the team which are: Author, Moderator, Reader, Recorder, and Inspector.

Author

Author is the person who originally constructed the work product. Author will be ultimately responsible for updating the work product after the inspection. Additionally, Author will address specific questions that arise concerning the content of the work product. Author also has to use their special knowledge to help identify and detect defects in the work product.

Moderator

Moderator is the leader of the inspection team. Moderator is responsible for ensuring that the inspection procedures are performed through out the entire inspection process. This includes ensuring team members perform their roles to the best of their ability. Below are the important issues that a Moderator responsible for:

- Verifying the work products readiness for inspection
- Verifying that the entry criteria is met
- Assembling an effective inspection team
- Keeping the inspection meeting on track
- Verifying that the exit criteria is met

Reader

The reader is responsible for leading the Inspection Team through the inspection meeting by reading aloud small logical units, paraphrasing where appropriate.

Recorder

The Recorder will document all defects that arise from the inspection meeting. Where the defect was found also will be documented. Recorder also assigned a defect category and type for each defect. All the defect data is recorded on a Software Inspection Defect List.

Inspector

Every member of the Inspection Team individuals play the Inspector role, independent of other roles assigned. The Inspector role is responsible for analyzing and detecting defects within the work product.

2.1.1.3 The Formal Process of Software Inspection

After Michael Fagan introduced software inspection, these processes is adopted with many variants. In this project, the software inspection model will mainly refer to model refined by Software Assurance Technology Centre, National Aeronautics and space Administration (NASA).

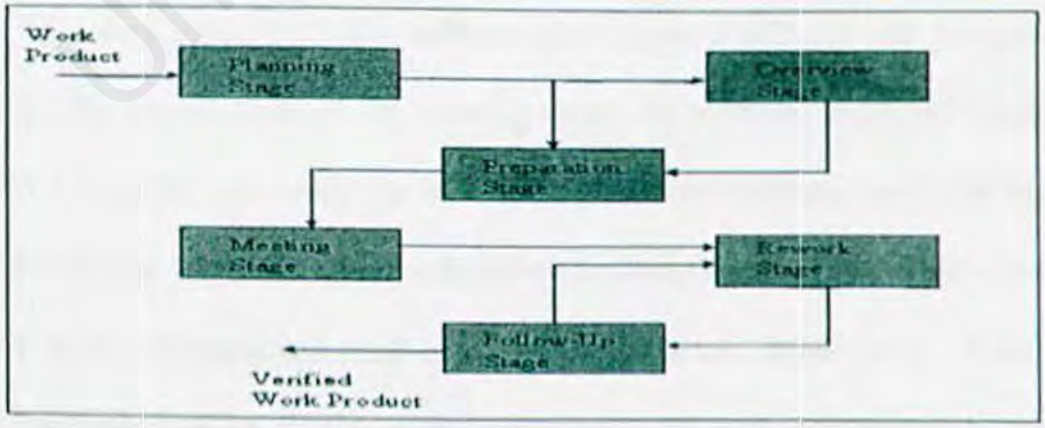


Figure 2.2 Software Inspection Model.

The software work product which to be inspected could be a Software Requirements Specification, High Level Design Document, or modules of code. But every product going to be inspected will go through the following process:

Table 2.1 Description of Software Inspection Process

Inspection Stage	Description
Planning	Identifies work product to be inspected and sets the inspection schedule.
Overview	Optional phase where team members who are unfamiliar with the work product to be inspected receive orientation.
Preparation	Team members inspect the work individually looking for defects in the work product.
Inspection Meeting	Inspection team members met to discuss possible defects in the work product.
Rework	The work product is revised to conform to requirements and specifications.
Follow up	The rework is verified, final inspection data is collected and summarized, and the inspection is officially closed.

The Planning Phase

When the author of the particular software work product believes that his work will satisfy all the entry criteria of the planning stage, the software inspection considered initiated. The author will notify the Moderator. Moderator will now verify the author's claim. Moderator will check whether this work product meet the entry criteria materials that are to be inspected and what condition the materials should be in. If the work product doesn't meet the criteria, work product has to be corrected to satisfy the criteria.

This early process is very important because a work product which is not in an expectable state is not worth for the effective use of the inspector's effort and time.

If the entry criteria are met, Moderator and Author will select the inspection team members. After that, the Moderator will determine whether an Overview meeting is needed or not based on the inspection members' information about the work product.

There are several guidelines that a Moderator should follow in this phase

- The work product that is being inspected should be frozen for the duration of the inspection process.
- There should be a minimum or 2 days lead time for the preparation.
- Maximum of 20 pages of the work product should be inspected at any time.
- The Author should not play the role of Moderator, Reader, or Recorder in an inspection.

The Overview Phase

This is an optional phase in the software inspection processes. To analyse and find out defects in a work product, the inspectors should have a certain level of knowledge of the work product. If Moderator thinks that the inspection team members lack this knowledge, then Moderator will decide to hold an overview meeting. This meeting is to educate and inform the inspectors. The Author will explain the work products functionality and provide a description of any techniques or representations that are used.

An overview is scheduled if one or more of the following circumstances apply:

- The inspection team is not familiar with the product.
- The product is new or is being inspected for the first time.
- Inspections are new to the project.
- Novel techniques have been used in the product.

The Preparation Phase

During this phase, each inspection team members inspect the work product individually to find out the defect. Moderator will assign a Defect Categories to each inspector to help them analyse the work product. Checklists should be used during this stage for guidance on typical types of defects to be found in the type of product being inspected.

Table 2.2 Checklist for Software Requirement Inspection

Defect Category	Key Questions
1. Clarity	Are the requirements clear and unambiguous?
2. Standards	Have all requirements standards been followed?
3. Completeness	Are all the requirements complete?
4. Level of Detail	Are the requirements free from design?
5. Consistency	Are requirements consistent? Are data structures and functions named and used consistently?
6. Functionality	Are functions correctly specified? Are inputs and outputs clearly specified? Are functions logically independent and highly cohesive?
7. Performance	Are the performance requirements for timing, memory,

	and resource utilization clearly defined?
8. Testability	Are requirements testable and verifiable?
9. Feasibility	Can each item be implemented with the available techniques, tools, resources, and personnel and with in the specified cost and schedule constraints?
10. Tractability	Are the requirements uniquely identified that they will be able to be traced back from design and analyse, and coding phase?
11. Modifiability	Are the requirements uniquely structured such that any necessary changes to the requirements can be made easily, completely, and consistently?

When an inspector analyses and studies the work product, all the potential defects is recorded but is not considered as defects yet. These defects will be discussed in the Software Inspection Meeting Stage for the determination to be considered defects of the work product.

The inspectors will recorded the potential defect found during this phase in the preparation log. This from will be sent to the Moderator before the inspection Meeting is kick off. Moderator will analyse the log carefully to determine that the team is adequately prepared for the Inspection Meeting. Else the Moderator should reschedule the date of Inspection Meeting because a not prepared team will waste time and not effective in finding defects. This log will return to the inspector during the Inspection Meeting for their use to pointing out defects and discussions.

The Inspection Meeting Phase

The Inspection Meeting phase gathers all the inspectors to find out the defects. The entire inspection team members attend this meeting. Moderator will lead this meeting and concentrates on finding the defect but not solving the defect. The Reader provides a logical reading and interpretation of the product, the Author provides clarifying information as needed, and the team identifies defects that are classified and recorded.

Following is the usual agenda during an inspection meeting

- Introduce the inspection meeting.
- Establish the preparedness of the inspectors.
- Read the work product, and identify and record the defects.
- Review the defects.
- Determine the work product disposition.

During the meeting, the agreed defects detected will be recorded in the Software Inspection Defect List. This Software Inspection Defect List is used to determine the disposition of the work product. The disposition referees to the procedure that should be used to verify the author's rework of the work product. Robert (Ebenau, 1994) suggests the following three disposition categories:

- A: Accept the work product as complete, without any further verification of rework. This does not require the work product to be totally defect-free, but it does require that there be no defects that cause the product to deviate from its

specifications, and that there are only a very few trivial defects that can be left to the Author's discretion.

- C: Conditionally accept the work product, with verification of the rework by the Moderator in review with the Author. In this case there are some major defects, but they are few, relative to the work product, and their rework is not expected to create any substantial changes in the "design premise" of the work product.
- R: Re-inspect the Author's rework. This disposition requires that the rework be examined by the Moderator, the Author, and at least one other member of the inspection team in a reiterating of the inspection meeting. For this case, there are a substantial number of major defects or rework that will change the original design premise of the work product.

The Rework Phase

The purpose of rework is to correct defects found during the inspection meeting. Author correct the work product refers to the Inspection Defect List. Author should correct all the major defects noted in the Inspection Defect List while minor defects should be resolved if cost and schedule permit. When all the defects are resolved, the Author will notify the Moderator to move to the Follow-up phase.

The Follow-up Phase

The Moderator will perform a formal verification on the revised work product. Maybe the Author included new defects inside during his attempt to correct the previously

identified defects. Robert (Ebenau, 1994) suggested that depending on the product disposition that was assigned during the inspection meeting, the Moderator will either:

- If product disposition was C, then the moderator will verify the revisions. The Moderator should focus on the content of the revisions as well as how the revisions interface with the rest of the document.
- If product disposition was R, then a full inspection process is performed but it is focused toward the revisions, their interdependencies.

After the Author has applied all requested changes and the Moderator has verified the revisions, then the remaining Inspection Summary Report are completed in by the Moderator.

2.1.1.4 Summary of Software Inspection

The following summarizes the essentials of the formal inspection process:

1. Inspections are carried out on products that have been completed by their author but not yet tested, reviewed, or otherwise approved or base lined.
2. The objective of the inspection process is to detect and remove defects. Typical defects are errors of documentation, logic, and function.
3. Inspections are carried out by peers of the author. Participants in the inspection process should represent organizations that will use or will be affected by the material being inspected.
4. Inspections should not be used as a tool to evaluate workers. Management is not to be present during inspections. When a management official has technical

expertise which is not available from other sources, that individual may be brought into the third hour.

5. A trained moderator leads inspections, and all participants should have training in the process.

6. Inspectors are assigned to and prepare for specific roles(e.g, reader, recorder, author).

7. Inspections are carried out in a prescribed series of steps from planning through follow-up.

8. Inspection meetings are limited to two hours.

9. Checklists of questions and typical defects are used to stimulate defect finding.

10. The product being inspected should be of an appropriate size that it can be inspected during a two hour meeting.

11. Correction of defects is the responsibility of the author, and is verified by the moderator. The inspection team must refrain from suggesting methods for correction during the inspection meeting.

12. Data and trends on the number of defects, the types of defects, and the time expended on inspections should be maintained. This information should be used to evaluate and improve the effectiveness of the inspection program.

2.1.2 Review of the Existing System

Although software inspection has been introduced for many years, but there are only several software inspection management on the market. Most commercial products only cover a little bit of requirements in software inspection process which is – defect tracking (which is in the preparation and inspection meeting phase). There are some better products which are developed in the purpose of software inspection research field but not for commercial selling.

Case Study 1:

Asynchronous/Synchronous Software Inspection Support Tool

Asynchronous/Synchronous Software Inspection Support Tool (ASSIST) is a generic tool designed to allow the enforcement and support of any inspection process. This is achieved with a custom-designed process modelling language (Inspection Process Definition Language, or IPDL), and a flexible document type system. ASSIST is based on a client/server architecture, where the server is used as a central repository of documents and other data. ASSIST supports both individual and group-based phases of inspection. Group-based phases can be performed synchronously or asynchronously, with the choice of same-place or different-place synchronous meetings.

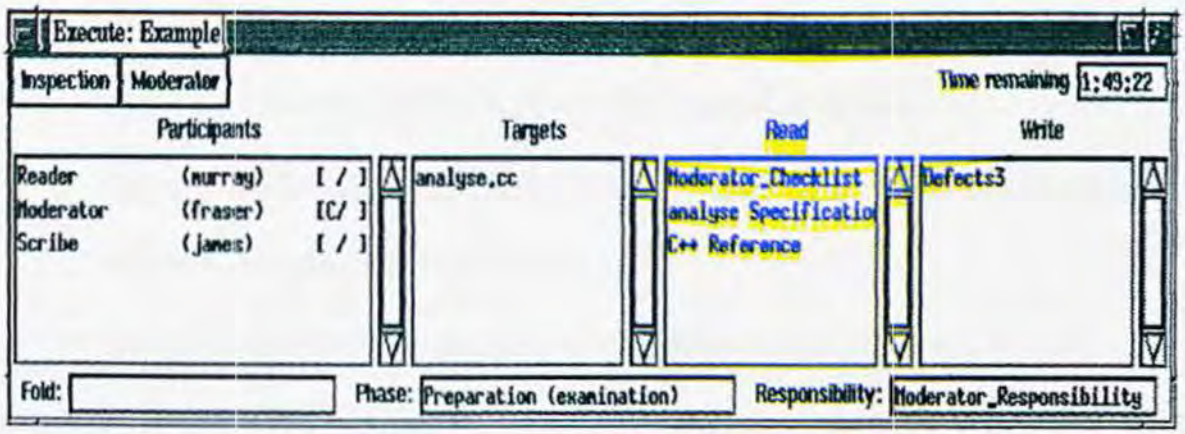


Figure 2.3 a sample interface of ASSIST

Comments:

- It is a research tool that uses a custom-designed process modelling language.
- It supports any inspection process and allows inspection of any type of document.
- It may be used to assist both group and individual inspections.

(Reference URL : http://www.cs.strath.ac.uk/~efocs/home/assist.html)

Case 2:

Rational ClearQuest

IBM Rational ClearQuest is a powerful and highly flexible defect and change tracking system that captures and manages all types of change requests throughout the development lifecycle, helping organizations quickly deliver higher quality software.

IBM Rational ClearQuest provides:

- Provides activity-based change and defect tracking.

- Manages all types of change requests, including defects, enhancements, issues and documentation changes with a flexible workflow process.
- Enables easy customization of defect and change request fields, processes, user interface, queries, charts and reports.
- Out-of-the-box provides predefined configurations and automatic e-mail notification and submission.
- Scales easily to support projects regardless of team size, location or platform.

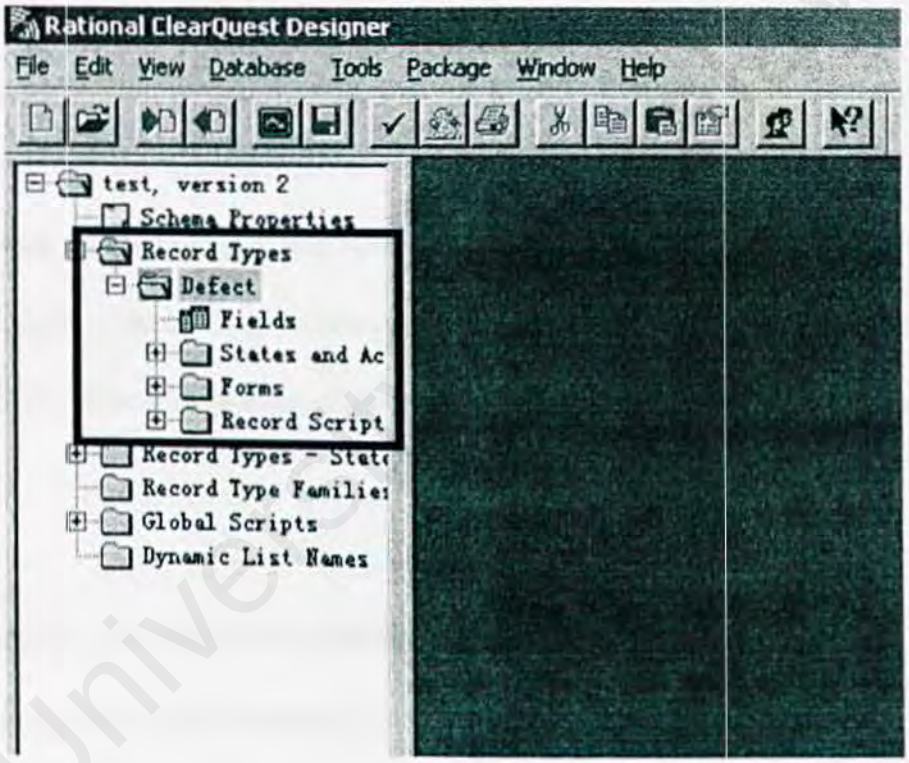


Figure 2.4 A sample interface of Rational ClearQuest Designer

Comments:

- There is one interesting feature of system where user can submit defects to ClearQuest via regular e-mail.

- ClearQuest provided an "email reader" to parse the information in the e-mail which has to be written in specific format; otherwise it will not function properly
- This system is mainly for defect tracking and only overlapped some parts of software inspection

(Reference URL : http://www-306.ibm.com/software/awdtools/clearquest/)

Case Study 3:

ReviewPro

ReviewPro is a proven, web-based Technical Review and Inspections solution. The software company claims that it will deliver a 10:1 Return On Investment (ROI). ReviewPro reduces the administrative overhead in planning and conducting reviews, shortens the overall review time, and increases the number of defects found early in the development cycle. The on-line, collaborative process removes issues of scheduling and allows distributed teams (customers & developers) to work together over the Internet.

Comments:

- ReviewPro is a web-based, groupware application.
- Users need only a web browser
- Total automation of the defect detection and removal method
- Architecture independent and can run on any web browser or messaging software.
- Compatible with any database used as it utilizes ODBC connectivity

- Primarily used for technical review process in the inspections in the software testing process
- It's a commercial software and licensed fees is not cheap
- It's a software recommended by senior software inspection expert—Tom Gilb

(Reference URL : http://www.sdtcorp.com/reviewpro.html)

Case Study 4:

Internet-Based Inspection System (IBIS)

Internet-Based Inspection System (IBIS) is software available under the IBIS License which is based on the MIT licence.

IBIS suggest that software inspection can be distributed across multiple sites if:

- It is supported by some Internet-mediated environment (basic requirements)
- Tool support as a web application with email-based event notification
- Inspection data stored and exchanged as XML documents

Comments:

- A web-based tool that aims to support geographically dispersed inspection teams.
- It adopts a reengineered inspection process to minimize synchronous activities and coordination problems.
- A research system by Filippo Lanubile from Italy

- IBIS is good from all areas except it is limited for inspection of requirement document only
- The inspection areas is not detail enough, limited to one checklist only

(Reference URL: http://cdg.di.uniba.it/research.html)

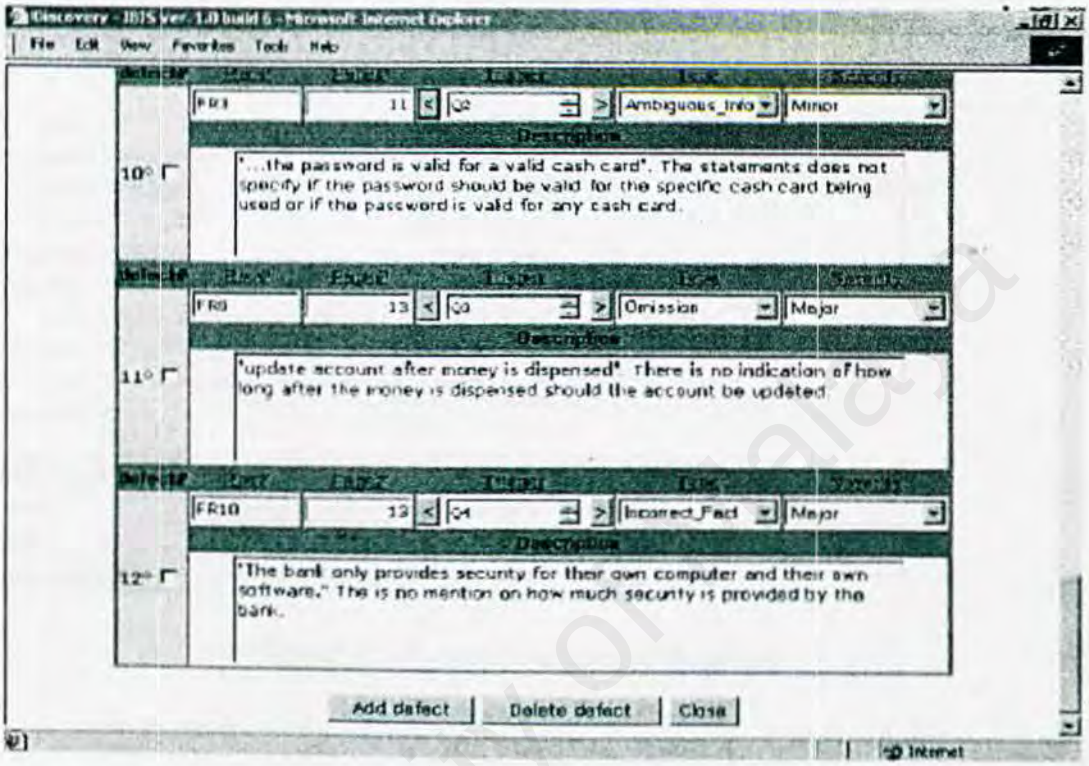


Figure 2.5 Sample interface of IBIS

Case Study 5:

BugZilla

Bugzilla is a "Defect Tracking System" or "Bug-Tracking System". Defect Tracking Systems allow individual or groups of developers to keep track of outstanding bugs in their product effectively. Most commercial defect-tracking software vendors charge enormous licensing fees. Despite being "free", Bugzilla has many features its expensive

counterpart's lack. Consequently, Bugzilla has quickly become a favorite of hundreds of organizations across the globe.

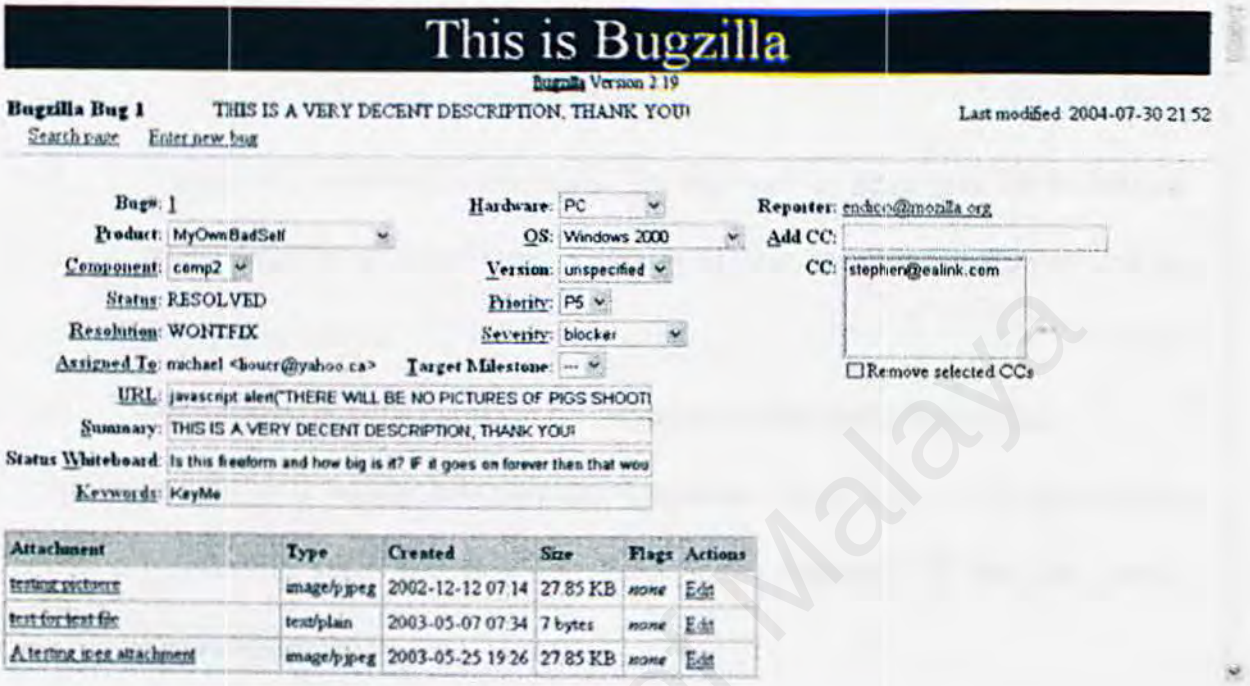


Figure 2.6 A sample interface BugZilla

Comments:

- Track bugs and code changes
- This actually a Linux based software tracking tool mainly used for bug tracking in development of Linux program.
- A code inspection tool.

(Reference URL: http://www.bugzilla.org/)

2.1.3 Proposed System

From what have been reviewed and analyzed, the proposed system will have features below:

i) System will provide the Moderator features such as determine the inspection goals, category of defect types, selecting of team members and severity to be detected in defects.

ii) System will provide checklist for requirement document inspections.

iii) System will supply template for Inspection Issue Log to be use during preparation stage and this will keep in the database for the use during inspection meeting.

iv) All the Inspection Issues Log will gathered together and discuss in the inspection meeting stage.

v) Inspection Logs will be processed and the redundant defects will be gathered together in a final list as they are high potential defects.

vi) An Inspection Summary Report is provided for Moderator at the end of inspection.

2.2 Technology Review

This subsection contains the review of technologies used so that developers can get a better understanding on the development tools that can be used to develop a project and also to get a better knowledge on the development methodologies used while developing a project.

2.2.1 System Architecture

System architecture exist to provide organizations flexible and robust infrastructure that, depending on how they are design, and cater the specify business needs. Below is description of different system architecture environments:

Client-Server Architecture

Client-server architecture is a distributed system model which shows how data and processing are distributed across a range of processors. In this approach, the system may be thought of as a set of services that provided to clients are treated differently in these system. The 'client' is allowed to request service from the server and the 'server' gives the services to clients. This approach has the ability to present the information to the user via a graphical user interface.

Advantage: shared data environment, ideal for small business

Disadvantage: software control, Software deployment and poor performance

Client-Server 2-tier

The two-tier architecture partitioned into two logical layers: The front-end and back-end. The front-end process is developed in a 4th Generation Language and allowed the user interacts on their personal computer. However, the back-end process is a database server and it is typically resides on a centralized server machine in a centralized environment.

Client-Server 3-tier

Client-Server 3 tier drive a logical division of the application components, the database component and the business rules that govern the processes the application supports. It provides an explicit layer for the business rules that sits between front-end and background. It allows encapsulates the business model associated with the system and separates if form the presentation and database code.

Table 2.3 Comparison between two client/server architecture

Feature	2-tier	3-tier
System Administration	Complex	Less Complex
Security	Low	High
Encapsulation of data	Poor	High
Performance	Poor	Good
Scale	Poor	Excellent
Application reuse	No	Excellent
Server-to-server infrastructure	Poor	Yes
Internet support	No	Excellent
Heterogeneous database support	Limited	Yes
Hardware architecture flexibility	Poor	Excellent
Availability	Poor	Excellent

2.2.2 Application Platform

Microsoft Windows XP Professional

Microsoft Windows XP Professional is the latest operating system released by Microsoft. Windows XP is built on Windows NT technology. It is easy-to-use, which have familiar Windows 98 user interface.

Its integrated Web capabilities and broad support for mobile computers and hardware devices makes it the easy way for business users to connect to the Internet anywhere and anytime. And its rock-solid reliability and improved manageability simplify desktop management for IT professionals.

The combined features of Windows XP Professional make it in the mainstream operating system for desktop and notebook computing in all organizations. It has the best business features of Windows 98 Plug and Play, easy-to-use user interface, and power management and made them better. It's also integrated the strengths of Windows NT standards-based security, manageability and reliability. Whether deploy Windows XP Professional on a single computer or via a worldwide network, Windows XP Professional increases the computing power while lowering the total cost of desktop ownership.

Advantages:

- More reliable, stable and easier to use. When a program freezes or crashes, the whole system isn't broke down.

- Nicer look of graphical user interface makes it user-friendly and provides a familiar interface for most users.
- Encrypting file system provides a high level of protection from hackers and data theft by transparently encrypting files with a randomly generated key.
- A new features called Remote Assistance lets you invite a technician to connect your PC to help solve problems
- Network friendliness XP includes simple wizards for setting up a home network, and it automatically recognized my speedy cable modem Internet service.

Disadvantages:

- Some of DOS based programs and drivers may not work under XP due to its true 32-bits coding.
- Some software and hardware won't operate under XP without updated patches or drivers. Extra spending is needed for XP-ready software and add-ons.

UNIX

UNIX is an operating system that originated at Bell labs in 1969 as iterative time-sharing system. It is very popular because of its large support base and distribution. UNIX is a multitasking, multi-user operating system. This means that there can be many people using one computer at the same time, running many different applications. (This differs from MS-DOS, where only one person can use the system at anyone time)

Advantages of UNIX:

- Powerful and motivate operating system.
- More flexible and can be installed on many different types of machines, including main-frame computers, supercomputers and micro-computers.
- More stable and does not go down as often as Windows does, therefore requires less administration and maintenance.
- Greater built-in security and permissions features than Windows.
- Possesses much greater processing power than Windows.
- The leader in serving the Web. About 90% of the Internet relies on Unix operating systems running on Apache, the world's most widely used Web server.

Disadvantages:

- It is too expensive to use because it needs a very powerful works station.
- Case sensitive
- Not user friendly, confusing for beginners and difficult to maintain
- Proprietary hardware is expensive

Linux

Linux is a free, UNIX-like operating system designed for Intel processors on PC architecture machines created by Linus Torvald. Linux is not UNIX, as UNIX is a copyrighted piece of software that demands license fees when any part of its source code is used. Linux was written from scratch to avoid license fees entirely, although the

operation of the Linux operating system is base entirely on UNIX and it shares UNIX's command set.

Linux supports a wide range of software, from teX (a text formatting language) to X (a graphical user interface) to the GNU C/C++ compilers to TCP/IP networking. It is well suited to function as a development environment for web applications. Its superior stability is a feature that cannot be beaten even by Windows. Linux is capable of running 24 hours 7 days a week without system failures or crashes. Memory management is dynamic and used memory is released after a particular application ends unlike Windows.

The famous Linux vendors are Red Hat, Mandrake and SUSE.

Advantages:

- It's free (Fees are charges from vendor for maintenance)
- It is stable and reliable
- It is a robust and complete enough to handle large tasks, as well as distributing computer needs
- It is developed under General public licenses (GNU) and its source code is freely available to everyone.

Disadvantages:

- Inherently unsafe because very malicious cracker in the universe has the source code to the website that developer under Linux
- It is developed by people world-wide then lack proper organized support.

2.2.3 Web Server

Web server is a software program to service hyper text mark-up language (HTTP) request. In addition, web servers also perform the following functions:

- Provide access control, determining who can access particular directories or files on the web server.
- Run scripts and external programs to either add functionality to web documents or provide real time access to database and other dynamic data. This is done through applications programming interface like CGI
- Enable management and administration of both the server functions and the content of the web sites.

Web server allows user to secure content over the internet using HyperText Transport Protocol (HTTP). The web server accepts request from web browsers like Internet Explorer and Netscape and then returns the appropriate HyperText Mark-up Language documents.

A survey carried out by NetCraft Survey on the date: **4th August 2004**, shows that Apache is the most famous web server on the Internet followed by Microsoft IIS and Netscape Enterprise (Based on response from 53,341,867 sites).

Table 2.4 Web Server Market share

Server	Market Share
Apache	67.70%
Microsoft-IIS	21.20%
Netscape-Enterprise	3.12%
Others	7.98%

Microsoft Internet Information Services (IIS)

Microsoft Internet Information Services (IIS) is the second most popular web server on the market. Microsoft released IIS with NT server as a free component turned for performance on the Intel platform. IIS is bundled in Microsoft Windows XP Professional as optional component. It is intended to run on a single platform. IIS is managed from the Administrative Tool under Control Panel in Windows XP. The IIS comes with three default services: WWW, FTP, and Gropher. Its Internet Service Manager application controls these services on this or any other IIS server on the network.

Advantages:

- Indexing, performance and security enhancement
- Well integrated server administration tools
- Easy to configure
- WebDav support makes for easier collaborative publishing

Disadvantages:

- No available for UNIX machine
- Documentation lacking on newest version

Apache

Apache is the high-end enterprise-level server for UNIX and windows platform. It is available free on the Internet. Apache web server reliably and quickly serves more than 60 percent of the current posted websites. Configuration manager for apache (Comanche) provides a quality cross-platform graphical tool for configuration and management of the apache web server and related software. Comanche run on most flavours of UNIX, Windows even on Apple Macintosh. With sufficient tools, Apache is easy in installation.

Advantages:

- Freeware
- Performance and robustness ,rock solid reliability and extensively
- Security
- Streamlined interface

Disadvantages:

- More extensive technical support requires the purchase of a third party support contract.

Netscape Enterprise Server

Netscape Enterprise Server is a high end enterprise level server for UNIX, AIX, HP-VX, IRIS, Solaris and Windows. The Enterprise Server is on a par with Microsoft IIS on Intel hardware and surpasses the Apache Server under dynamic testing. The server can be configured through editing of the configuration text file. The newly reengineered Web User Interface (WUI) has made it even easier to configure than Microsoft IIS. The WUI can be used to set up the server, either from a Microsoft Internet Explorer or Netscape Navigator browser. Besides that, an inexperienced user will very pleased for the well-designed context-sensitive online help.

Advantages:

- Wide spread platform support
- centralized server management
- Integrated search engine
- SMTP support
- End user publishing capabilities

Disadvantages:

- Price
- Complexity

2.2.4 Web Programming Language

ASP.NET (Advance Active Server Page)

ASP.NET is the replacement for Active Server Page (ASP), offers compiled web pages which will make the processing of web request much more efficient. ASP.NET is the new hosting environment that enables developers to use the .NET Framework to target Web-based applications. Actually, ASP.NET is more than just a runtime host. ASP.NET is a complete architecture for developing websites and Internet distributed objects using managed code.

A key feature of ASP.NET is the separation of code into a separate file from the HyperText Mark-up Language (HTML) page that calls it. This 'code behind' concept helps clarify the roles of designers and developers, and neatly accommodates another important .NET technology, namely XML Web Service.

Advantages:

i) Easy Programming Model

Building of real world web application will dramatically easy by using ASP.NET. Fewer codes are needed compared with classic ASP. The ASP.NET page works on all browsers include Netscape, Opera, AOL, and Internet Explorer.

ii) Flexible Language Options

Unlike classic ASP which support only interpreted VBScript and Jscript, ASP.NET can support various type of .NET language including built-in support for VB.NET, C# and Jscript.NET.

iii) Great Tool Support

Any text editor can be use to write ASP.NET webpage, even using Notepad. Visual Studio .NET adds the productivity of Visual Basic-style development to the web. Visual Studio .NET also provides integrated support for debugging and deploying ASP.NET Web applications.

iv) XML Web Services

XML Web services allow applications to communicate and share data over the Internet, regardless of operating system or programming language. ASP.NET makes exposing and calling XML Web Services simple. Any class can be converted into an XML Web Service with just a few lines of code, and can be called by any SOAP client.

Java Server Page (JSP)

Java Server Page (JSP) is a technology based on the Java language and provides a simplified, fast way to create web pages that display dynamically-generated content. It is a technology that lets users mix regular, static HTML with dynamically-generated HTML. JSP was developed by Sun Microsystems to allow users create dynamic web based content using server-side (middle-tier) processing. The JSP specification defines the interaction between the server and the JSP page, and describes the format and syntax of the page. JSP simplifies the process of creating the dynamic pages by separating the application logic in portable, reusable Java components. Additionally, it also simplifies the task of building web applications that work with a wide variety of web servers, browsers and development tools.

JSP pages are not restricted to any specific platform or to any web server. JSP pages encapsulate the logic that generates the content for the page by using XML tags and scriptlets written in the Java programming language. It passes any formatting tags such as HTML or XML, directly back to the response page. As such, JSP pages separate the page logic from its design and display.

JSP technology builds on the strength of the Java technology family and the multi vendor Java community. It extends the core capabilities of the Java platform to create powerful, flexible, and easy-to-maintain dynamic web pages.

2.2.5 Authoring Tools

Microsoft Visual Studio.NET 2003

Microsoft Visual Studio.NET 2003 is the most powerful yet Integrated Development Environment (IDE) tools released by Microsoft to battle Java. Visual Studio.NET is a full featured IDE with excellent cross language support that can create web-based application easily. Despite producing programs for only one platform, Windows, Visual Studio offers the widest range of languages: Besides C++, C#, and Visual Basic, it includes a .NET version of Java called J#.). In addition, via the .NET Common Language Runtime (CLR), developers can incorporate code from legacy languages such as COBOL and FORTRAN. As a result, languages can be mixed and match besides can share components of each one within a single project.

Visual Studio.NET comes with Visio, which provides excellent Unified Mark-up Language (UML) diagramming. Class stubs can be generated from diagrams to start with enterprise projects.

Visual Studio offers wizards to get developers started with projects and components. It has the ability to generate ADO.NET datasets (for modeling database tables and views) and Web forms, including master/detail relationships.

Design tools for creating Web forms (using ASP.NET) and Windows forms (for standalone applications) are excellent. The Visual Basic-like drag-drog-double-click is accessible to most developers.

. Advantages:

- It offers multiple language support
- It has a rich set of libraries
- It's open standard friendly (HTTP and XML)
- It's code is compiled natively, regardless, of language or deployment (web or desktop)

Disadvantages:

- It's yet another platform to consider, which generally means rewriting and learning new skills
- Microsoft tends to have good ideas, but mediocre implementation
- Only available on Windows

Macromedia Dreamweaver MX

Macromedia Dreamweaver MX is the solution for professional web site design and production. It is a single integrated environment to create, build, and manage websites and Internet applications. Macromedia Dreamweaver MX combines its renowned visual layout tools with the rapid web application development features of Macromedia Dreamweaver UltraDev and the extensive code-editing support of Macromedia HomeSite, enables the web developers to take advantages of the latest web technologies, including .NET, ColdFusion MX, PHP, XML and XHTML.

Advantages:

- Macromedia Dreamweaver MX does have a clean up HTML command for files created in other programs, such as FrontPage.
- Fireworks is web-graphic software meanwhile Flash is web-animation software. Both are from Macromedia.
- It generates clean HTML that is easy to customize
- It has useful interface items. Enable view in full design, full code, or half design or half code
- Comes with complete HTML Debugger

Disadvantages:

- No back-end programming available for form submission
- Too much confusing panels

2.2.6 Database Management System

Microsoft SQL Server 2000

Microsoft's leading windows database and data warehousing. It has highly scalable and high performance relational database management system. It can manage a large amount of data in all multi-use distributed client server environments. It is design to understand Structured Query Language (SQL).

It also provides the enterprise data management platform which can adapt quickly in a fast changing environment.

Advantages:

i) Fully web enable

- rich integrated xml support
- powerful search capability
- web enable analysis
- web access to data
- ensure the application are secure at any network environment

ii) Highly scalable and reliable

- Highly availability
- Scalability
- Large memory support
- Gain performance from existing hardware by storing query results.

iii) Faster time to market

- Simplified database administration
- Improved developer productivity
- Ease of installation, deployment and use

Oracle

Oracle has distinction of being the first company to create and sell a commercial RDBMS that used SQL. Oracle is the most popular database management system or rather relational database management system. The most powerful feature for oracle is portability and scalability. Oracle's architecture supports large databases and large number of concurrent user. It enforces data integrity and provides file safe security features to limit and monitor data access. It also has high translation rates and availability. Oracle is an open system and it adheres to industry which is accepted standard for data access language, operating system, user interfaces and network communication protocol

Advantages:

i) transaction processing

- Data partitioning
- Material views
- Query optimization

ii) Reliability

- Complete data protection
- Online data evaluation
- Self service error correction

iii) Security

- Single sign-on oracle advance security
- Selective data encryption
- Secure data sharing

iv) Data warehousing

- Data mining
- Data warehousing Extraction-Transformation-Loading (ETL)

v) Managing tools

- Intelligent self managing and turning
- Manage the entire stack
- Pin point diagnostics
- Database resource management

MySQL

MySQL is the most popular, free, open source database management system for the UNIX/LINUX platform. It contains an enormous amount of functionality and power. Using MySQL, a complex set of database and tables can be developed by using a simple set of commands for inserting, retrieving, deleting and updating data. MySQL can be accessed and manipulated from a huge number of popular programming languages. It contains built-in support for every common filed type. it also supports a subset of advanced querying and grouping functions. MySQL allows per-server password allocation which improved the security.

Advantages:

- MySQL is free download and comes complete with all tools that need to get start
- MySQL is completely optimized for both UNIX and Win32 platform. it uses in-memory hash tables, thread-based memory allocation and kernel threads that are capable of utilizing multiple processors, an highly optimized individual pre-compiled class libraries
- MySQL supports a variety of connection methods, for example unit socket and TCP/IP sockets

2.2.7 Data Access Technology

Active Data Object for .NET (ADO.NET)

ADO.NET is Microsoft's latest data access technology and, a part of the .NET Framework. ADO.NET is essentially a collection of classes that expose methods and attributes used to manage communications between an application and a data store. ADO.NET provides an extensive set of .NET classes that facilitate efficient access to data from a large variety of sources. It also enables sophisticated manipulation and sorting of data. Besides that, it can forms an important framework within which to implement inter-application communication and XML Web Services. ADO.NET can be used in any consumer application that needs to connect and communicate with, data sources such as Microsoft SQL Server as well as data sources exposed via OLE DB and XML.

Advantages:

- Interoperability - All data in ADO.NET is transported in XML format, meaning it's simply a structured text document that can be read by anyone on any platform.
- Scalability - ADO.NET promotes the use of disconnected datasets, with automatic connection pooling bundled as part of the package.
- Productivity - ADO.NET able to improve overall development time.
- Performance - Because ADO.NET is mainly about disconnected datasets, the database server is no longer a bottleneck and hence applications should incur a performance boost.

Chapter 3 Methodology

A methodology may be defined as a collection of procedures, techniques, tools and documentation aids. The purpose of methodology is to explore the processes of developing software and save the time in system development. Each methodology has it own distinct objective, if followed, can lead to the development of an application.

Choosing an effective and ideal development methodology is very important in order to make the project development done on time and reduce the risk. In software engineering, there are many type of software processes model which can be use to develop a system, such as waterfall model, prototype model, V model, spiral model, rational unified process and so on. There is no exactly a right way to develop a system, each development method had it own strength, advantages and suitability, depending on the situations, the way they are applied and who is involved in the development process. If the developer chooses the not ideal or not suitable method, then it will reduce the quality, presentation of the system.

From research, two most common paradigms in software development are structured approach and object-oriented approach. The methodology for InspectionManager is object-oriented approach with Rational Unified Process (RUP).

3.1 System Development Life Cycle

System (software) development generally takes the form of a life cycle. This life cycle called system development life cycle (SDLC). All systems go through the same generic stages in their lifetime. The stages are shown in the figure 3.1 below:

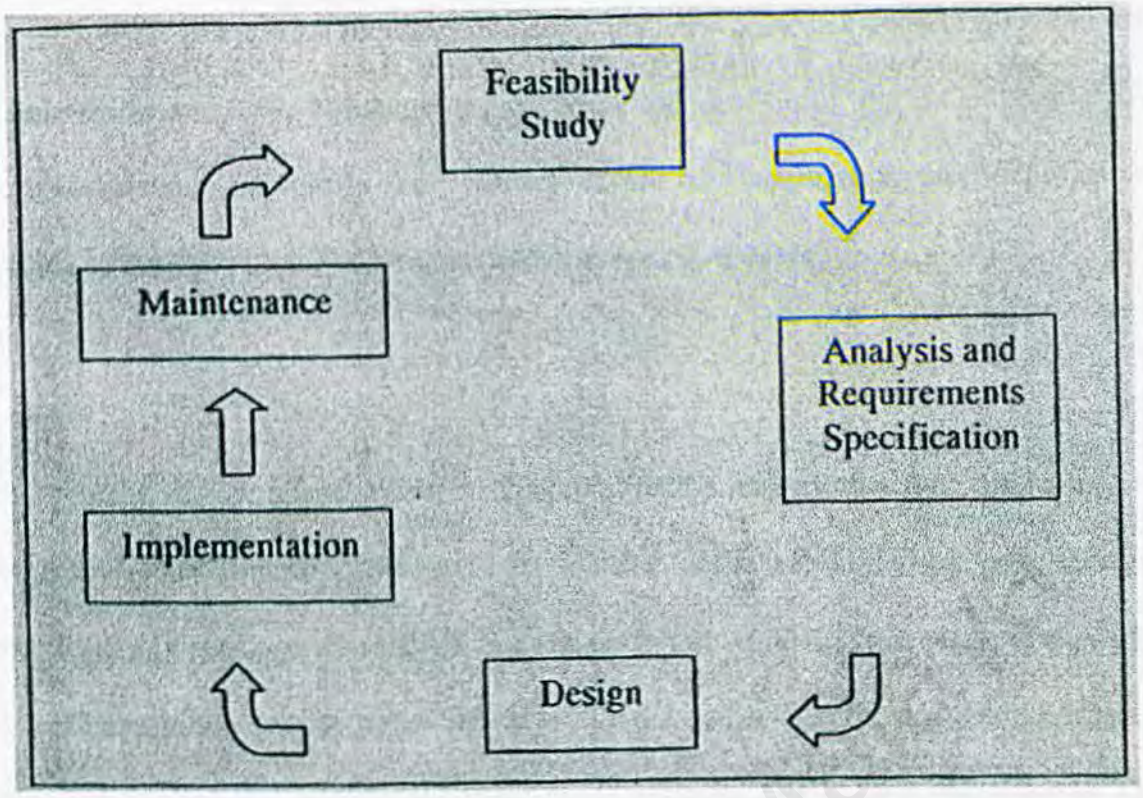


Figure3.1: System Development Life Cycle (SDLC) Stages

Feasibility Study

This phase begins with a problem or desired change in an information system or business process. A preliminary investigation is then performed to clearly identify them. This step is crucial because if the business' needs are not correctly identified the end product will most certainly miss the mark. The end result of this phase is a report that illustrates business considerations with the use of cost-benefit analysis and a proposal that is based on economic, technical and operational aspects.

Analysis and Requirements Specification

The main purpose of the systems analysis phase is to begin with a clear understanding of business requirements and build a logical, non-physical model of the new system. This phase also includes the sub steps of requirements, data, process and object modeling.

The end result is a system requirements document that describes management and user requirements and also Contains an overview of the entire project, the analyst's recommendation, as well as a recognition of the best alternatives that identifies the various strategies with the advantages and disadvantages of each.

Design

The systems design phase creates a blueprint for the new system that will satisfy all documented requirements, whether the system is being developed in- house or purchased as a package. The model that was built during the analysis phase is expanded in detail to include interfaces, the type of data input and output, the processes performed as well as backup and security measures. Once again the end result is documented. This time it comes in the form of a systems design specification report that is submitted to management and users for their approval.

Implementation

In the systems implementation phase the new system is constructed, programs are written, documented, and tested, and the system is installed. If the system was purchased as a package the analysts carry out any needed modifications and configure the system for use. The goal of this phase is to present a fully functioning and documented system, regardless of its origin, to train new users on and transition all of the old system's duties upon.

Maintenance

In the final phase, the IT staffs maintain and enhance the system. The objective during this phase is to maximize return on the IT investment. Analysis is also very important during this phase. System performance and operation costs must be continually monitored to determine if the new system is really a worthwhile replacement.

3.1.1 Structured Analysis and Design Methodology

Structured Analysis and Design methodology (SSADM) was developed by Ed Yourdon in the early 1970's. Its principles focus on the analysis and design phases of the life cycle and allow the flexibility to return to earlier phases when necessary. This in turn reduces maintenance time, effort and expense.

Structured Analysis and Design is a type of structured methodology that based on modelling. In structured methodology, the whole system is view as a single function. This system is later decomposed into set of functions which will then be grouped according to hierarchy. It is actually a decomposition approach where a System is decomposed into several smaller modules.

The main aspect of Structured Analysis and Design methodology philosophy are user involvement, the three views, top-down approach and separation of logical and physical methods. This methodology covers most of the system development life cycle (SDLC) from the feasibility study to system design. The strategy planning is part of Structured Analysis and Design methodology and it is to be carried out before a project is undertaken. This methodology also does not cover implementation maintenance phases

of the SDLC. However, it makes both implementation and maintenance easier by providing accurate documentation of the system. Structured Analysis and Design methodology is based on the following hierarchy of activities module, stage, step and task.

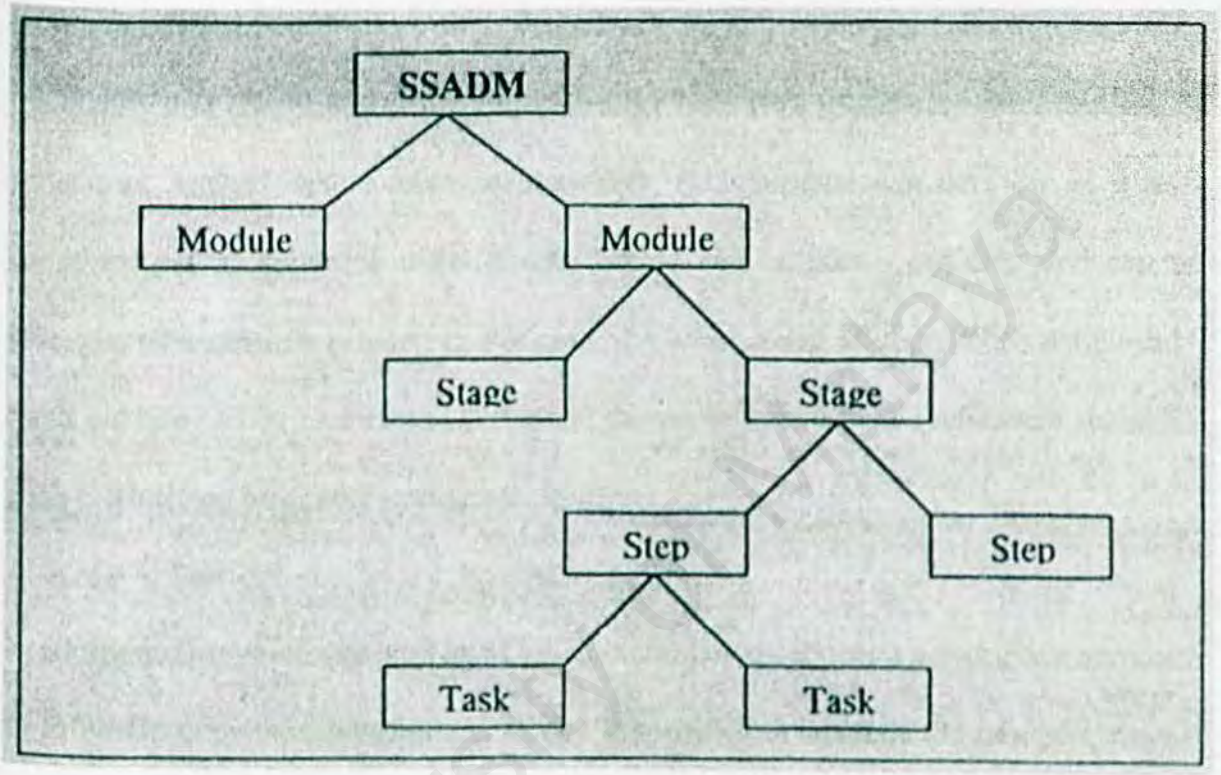


Figure 3.2 Hierarchies of Activities

There are five modules in this methodology:

- Feasibility analysis
- Requirements analysis
- Requirements specification
- Logical system specification
- Physical design

Every stage is divided into steps and each set is presented as a set of tasks, all of which are numbered for easy identification. Moreover, every step is thoroughly described both in terms of the activities that are to be carried as well as products, which are the result of the activity. Each step is associated with a number of techniques and tools, which the developer should use to model the system. Structured Analysis and Design methodology is thus product-driven Project management will be largely concerned with monitoring the quality and completeness of products rather than with monitoring the activities. In Structured Analysis and Design methodology, each module can be view as a self-contained set of activities with its own inputs and outputs / products and can be managed as a separate project. In this case, the consistency of approach is established. Each module can be understood better and the achievement of the milestone stated for each module can be clearly evaluated.

In addition, Structured Analysis and Design methodology adopts a prescriptive approach to information system development in that it specifies in advance the modules, stages, steps and tasks that have to be carried out, the deliverables produced and the techniques involved. Thus, SSADM can helps to overcome the chance that requirements are misunderstood. Besides that, this methodology also helps to overcome the problems concerning limited user involvement and the usage of inadequate analysis and design tools and techniques.

3.1.2 Object-oriented Methodology (OOM)

Object-oriented methodology is an approach to software design based on objects and asses. The abstraction concept forms the basis of object-orientation. An object is simply a real-world thing or concept that has an identity, state (describe the object (attributes) and the value of those attributes), and behaviour (represented by functions (methods) that are used or change the values of the object's attributes) An object is an instance of a class A class describes a set of objects with common attributes and behaviours.

3.2 Methodology Consideration

For InspectionManager, the object oriented approach is being adopted instead of structured approach because there are several advantages of using the object oriented methodology in Software development such as:

a) Higher level abstraction

The object-oriented approach supports abstraction at the object level because object encapsulated its data and functionalities.

b) Seamless transition among different phases of software development

The object-oriented approach uses the same language in analysis, design, programming and database design. This approach reduces complexity and redundancy and makes E-shopping system more robust and less error prone.

c) Encouragement of good programming technique

The object-oriented approach raised the level of abstraction from the function level to the object level and this helps to promote clearer designs with easier implementation and offers better overall communication in the InspectionManager.

d) Promotion of reusability

In object-oriented approach, objects are reusable because they are modelled directly from the real-world problem domain. Object is discrete and it stands on itself or within a small circle group (other objects). Therefore objects can be kept in an object library for future use to develop E-shopping system.

The Rational Unified Process (RUP)

The Rational Unified Process (RUP) is a software design methodology created by the Rational Software Corporation. It captures many of the best practices in modern software development in a form that is suitable for a wide range of projects and organizations. It embeds object-oriented techniques and uses the UML as the principal notation for the several models that are built during the development. The RUP is also an open process framework that allows software organizations to tailor the process to their specific need, and to capture their own specific process know-how in the form of process components. (Philippe Kruchen, 2000)

Rational Unified Process (RUP) used in the object-oriented software development methodology for better understanding of object-oriented concepts and provides a clearer

overall view of object-oriented system development. It is indeed very unique, compared to other software development process in term of completeness. Other software development process models (such as Waterfall model and Spiral) normally only specify what phases are involved in a software development process and the interactions among theses phases.

RUP is a complete process framework that includes every element in a software development process such as phases, workflows, interactions among them, milestone of each phase, activities within each workflow and artefacts or deliverables of each workflow.

Another difference between RUP and other software development process is that other software development processes only involve phases (these phases are known as workflows in RUP) within the development process. RUP have involves both phases and workflows. The five workflows cut across the four phases and these workflows are iterated within each phase to achieve a certain major milestone of the phase. The completion of each iteration on the other had marks a minor milestone.

Characteristics of the Rational Unified Process

The RUP is an iterative process. An iterative approach advocates an increasing understanding of the problem through successive refinements and an incremental growth of an effective solution over multiple cycles. Built into iterative approach is the

flexibility to accommodate new requirements or tactical changes in business objectives. It also allows the project to identify and resolve risks sooner rather than later.

The RUP's activities emphasize the creation and maintenance of models rather than paper documents. Models, especially those specified using the UML provide semantically rich representations of the software system under development. The purpose of focusing on models is to minimize the overhead associated with generating and maintaining documents and to maximize the relevant information content.

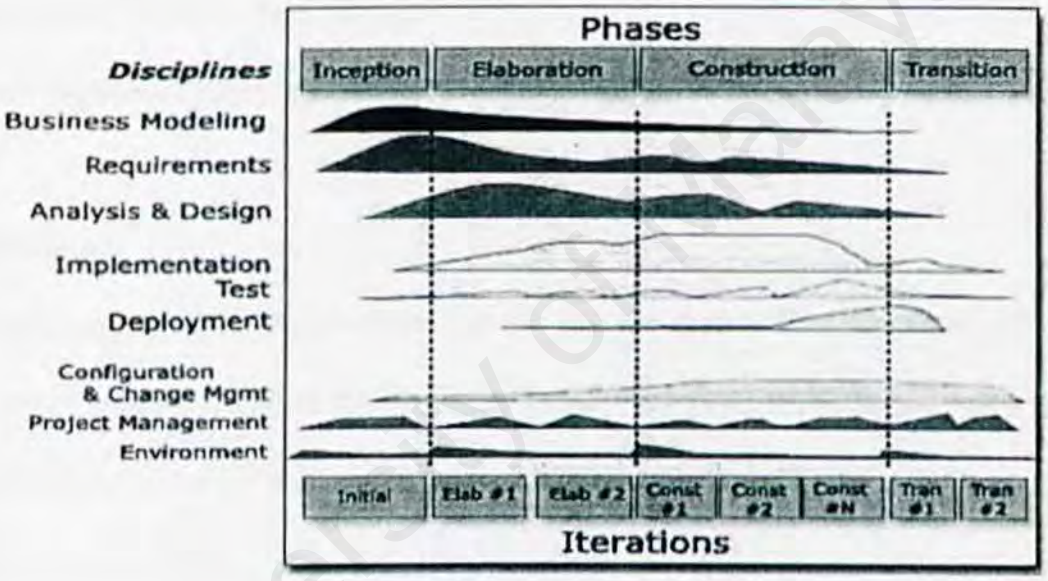


Figure 3.3 Four Phases of Rational Unified Process

Rational Unified Process consists of the following four phases:

Inception phase: establish the business case for the project

Elaboration phase: establish a project plan and a sound architecture

Construction phase: grow the system

Transition phase: supply the system to its end-users

Inception:

Primary goal: Defines the scope of the propose project, and the business case for the InspectionManager. The elements identified in the inception phase are used to establish the viability of the suggested system. Several changes have been made to the methodology so that it is suitable for this project. Task performs include:

- Identifies the requirements and scopes of the InspectionManager.
- Outlines the InspectionManager's architecture.
- Basic use case diagram of InspectionManager (20%).
- Project plan for InspectionManager.

- Major milestone: InspectionManager's life-cycle objectives.

Elaboration:

Primary goal: Analyze the problems domain and the system's architectural outline for InspectionManager. Analyze the ability to build the InspectionManager based on the constraints identified (schedule constraints, resources constraints, etc).

Task performs include:

- Capture the functional requirements for InspectionManager.
- Outline the software architecture description for InspectionManager.
- Use case diagram for InspectionManager (80% complete).
- Sequence diagrams for critical tasks.
- Prototype for some functions.

-Major milestone: InspectionManager's life-cycle architecture.

Construction:

Primary goal: Detail design of the InspectionManager as well as the corresponding source code. Task performs include:

• Building the InspectionManager iteratively and incrementally to ensure the system meets the system requirements

• Construct the user manual to provide help option.

Major milestone: Initial Operational Capability of InspectionManager.

Transition:

Primary goal: Final release of Component-based E-shopping application to customers.

Task performs:

• Refine the InspectionManager to overcome the unidentified problems.

Train customers and maintainers on operating the InspectionManager.

Major milestone: Release of InspectionManager.

Five Workflows of Rational Unified Process

There are five workflows in Rational Unified Process that cut across all the four phases. Workflow is a set of activities that various project workers perform. Below shows the workflows and models that are used in Unified Process:

a) Requirements Workflow

The Requirements workflow's activities focus on the building of the use case model. The use case model captures the functional requirements of the system and it plays an important role as the foundation from all the development work.

b) Analysis Workflow

Analysis workflow's activities are centered towards the building of the analysis model, which helps to refine and structure the functional requirements captured by the use case model.

c) Design Workflow

The main activities of the Design workflow are directed at building the design model, which describes the physical realizations of the use cases in the use case model as well as the contents of the analysis model. The design model acts as an abstraction of the implementation model.

d) Implementation Workflow

The main activities of the Implementation workflow are channeled towards building the implementation model, which describes the way that the design model's elements are packaged into software components.

e) Test Workflow

The key activities of the Test workflow are targeted at building the test model, which describes how the executable components from the implementation model being used during the integration and system test. (Jacobson, 2002)

3.3 Information Gathering

Fact finding techniques are essential skill for all system analysts. It is a classical set of techniques used to collect information about system problems, opportunity, system requirement and priorities.

The methods used in gathering the required for developing system are:

Internet Surfing

The internet is a platform where a lot of information can be acquired. With development of search engines such as Googles, Yahoo and Lycos, some ideas from similar system and some interesting web design can be collected. Besides, a lot of information on this system, development tools and technologies, database, programming languages, project methodology, and also client-server computing knowledge can be acquired through Internet. Below are the two main reference from Internet:

- National Aeronautics and Space Administration. *Software Formal Inspections Guidebook*, NASA-GB-A302. August, 1993.
- National Aeronautics and Space Administration, Software Formal Inspections Standard, NASA-STD-2202-93, April 1993

Book and Reference

Book and reference are used to get the information that needed to develop the system. Besides that the library services such as online e-book to get references. The main online e-book and research topic from other is the Association for Computing

Machinery (ACM) Portal. URL: *http://portal.acm.org/portal.cfm* . There are a lot of references that contains here such as programming, database, development tools and web design. All references are very useful in developing this system.

The main book use for the project research is:

- Gilb,T. and D.Graham (1993) *Software Inspection*. New York: Addison-Wesley Pub Co.

Further more, other source of reference is senior's thesis. From senior's thesis, clear ideas about the report format and content for this thesis report can be acquired. Refer to the thesis at FSKTM document room where got many thesis is collected in past few years, a lot of information is elicited.

Discussion with supervisor and friends

A discussion with supervisor and friends has been practiced to get help and advices during the development of the project.

Chapter 4 System Analysis

System analysis is an essential and important phase in software life cycle. The main purpose of this phase is to determine clearly of all necessary requirement before proceeding into subsequent phase. In shorts, system design is the process of gathering and interpreting facts, diagnosing problems, and using the information to recommend improvement to the system.

4.1 System Requirement Analysis

System requirement analysis covers the area of functional and non-functional requirement of the proposed system – InspectionManager. It is an important feature of the system that makes the system capable to proceed while fulfilling the system's intention.

4.1.1 Functional Requirements

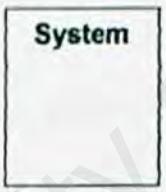
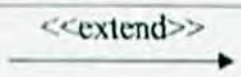
A functional requirement is a function or service that must be included in the system to satisfy the business need. Functional requirements state what the system should provide, how system should react to the particular situation and so on.

Use Case Diagram

A use case diagram is central to modeling the behavior of a system, a sub system, or a class where it shows a set of use cases, actors (a special kind of class) and their relationships. Use case diagrams are essential for visualizing, specifying, and

documenting the behaviour of an element, for testing executable systems through forward engineering, and for comprehending executable system through reverse engineering. Table 4.1 show the notation used in a use case diagram.

Table 4-1: Notation used in a Use Case Diagram

Elements / Relationships	**Notation**	**Description**
Use case		a sequence of actions that provide something of measurable value to an actor
Actor		a person, organization, or external system that plays a role in one or more interactions with the system
System Boundary		to indicates the scope of the system where anything within the box represents functionality that is in scope and anything outside the box is not.
Association		lines connecting use cases and actors to one another, with an optional arrowhead on one end of the line
Extend		The base use case implicitly incorporates the behaviou of another use case at a location specified indirectly by the extending use case

The use case below shoes the functional requirement of InspectionManager

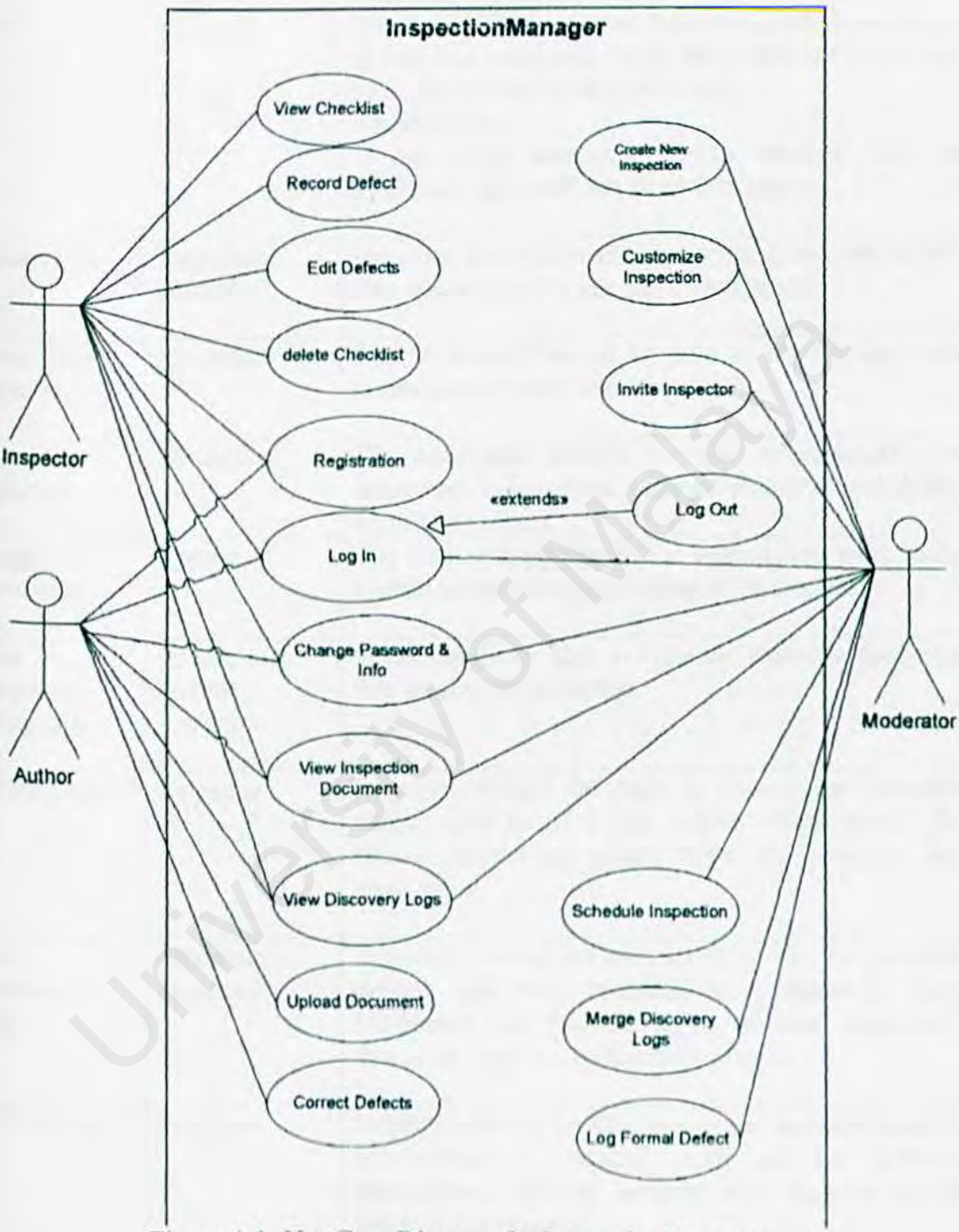


Figure 4-1: Use Case Diagram for InspectionManager

Description for InspectionManager Use Case Diagram

Table 4-2: Description for InspectionManager Use Case Diagram

Use Case	Actor	Description
Log –in	Moderator, Author, Inspector	This function checks for the level of the user when login to the system and it will allow the user to the privileges that they have right to access. User needs to key in a valid user name and password to be able to access the personal information. Alternate flow: If not being authorized, error message will be displayed. User will not be able to log in
Registration	Inspector, Author	Inspector and author must be a registered user before they can proceed in the inspection process.
Create New Inspection	Moderator	The Moderator should be able to register here and create a new inspection
Customize Inspection	Moderator	The Moderator should be able to customize the inspection information such as objective and defect types to be detect.
Upload Document	Author	The Author should be able to upload their work to the system as the software product to be inspected.
View Inspection Document	Moderator, Author, Inspector	Users should be able to view the software document that need to be inspected.
Record Defect	Inspector	Inspector should be able to record the potential defects they found in the system which specify the source document, page, type of severity and description.
View Discovery Logs	Moderator, Inspector	Inspector should be able to view all the potential defects that they recorded in a discovery logs. Moderator also has the rights to view inspector's discovery logs to validate their efforts.
Edit Defects	Inspector	Inspector should be able to edit the potential defect's information if needed such as the defect's description, type of severity and location in the inspected document.
Delete Defects	Inspector	Inspector should be able to delete each of the potential defect's information if needed.

Change password and information	Moderator, Author, Inspector	Users should be able to change their own password and information if needed.
View Checklists	Inspector	System should be able to let the Inspector view the checklists as a guideline for Inspector during the finding of potential defects.
Correct Defects	Author	Author should be able to record down the corrections that made for a particular defect such as correction description and Author's comments.
Schedule Inspection	Moderator	The Moderator should be able to customize the inspection process such as the meeting date, inspection overview date, and specified the documents to be inspected.
Invite Inspector	Moderator	Moderator can act as the power user to invite a group of Inspector to form a inspection team.
Merge Discovery Logs	Moderator	The Moderator should be able to merge all similar defects discovered by various Inspectors into a merged log for use during Inspection Meeting.
Log Formal Defect	Moderator	The Moderator should be able to log the correct/agreed defect into the formal Defect List.
Log out	Moderator, Author, Inspector	Users should be able to log out after finishing their task.

4.1.2 Non-Functional Requirements

A non-functional Requirement is a description of features, characteristic and attributes of the system. It also decrypts the constraints that may limit the boundaries of the proposed system. Below are some of the non-functional requirements for InspectionManager.

- **Accuracy and Consistency**

  The system should return the correct and complete set of information each time as requested by the user.

- **Reliability**

  The system should make the correct respond and provide error handling ability. It should be reliable in performing its daily functions and operations. For example each time the 'save' button is snapped, it must store the data into the right database of file.

- **Security**

  The system should be equipped with efficient security. Security in the system will be implemented by using user password and authentication manner. Information in the database should only be viewed, created or modified by the rightful user.

- **Simplicity and User friendly**

  The system by itself should be made simple. Users will not need to know so much about how the system goes. They should only be made to understand what they need to do when they want to generate a function. Moreover, usage of

suitable and meaningful caption will help the user to consume the system easier. Therefore, an attractive and easily understand user interface is needed.

- **Multi-user environment**

  The system will be implemented as a web based system and services. As it is a web based system, there are possibilities that there will be more than one user trying to access the system at the same time frame. Therefore, the system should be able to handle this situation.

- **Manageability and Maintainability**

  The module within the system should be easy to manage. The system can be corrected whenever an error occurs. Furthermore, it could be adapt to new demand and requirement or enhanced in the future.

- **Expandability**

  The system should be able to be extended to accommodate more functionality in the future.

4.2 Hardware and Software Requirements

Hardware and software requirements describe the constraints on computers and peripheral equipments. Hardware and software requirements need to be decided to determine the performance requirements' feasibility. Both hardware and software requirements are divided into runtime and development requirements.

Table 4-3: Hardware and Software Requirements

	Run Time	Development
Hardware Requirements	• 233 MHz Pentium / higher microprocessor / or equivalent • Random Access Memory : 64 MB and above (128MB recommended) • Hard disk : 2.5 GB and above • Standard input and output • Others standard computer peripherals	• Pentium IV 1.8 Gigabyte • Random Access Memory : 512 MB • Hard disk : 40GB • Display : GeForce 2 display card • Others standard computer peripherals
Software Requirements	• Windows 2000 server (Windows XP is recommended) • Any web browser (Internet Explorer 5.5 is recommended)	• Windows XP • Internet Explorer 6.0 • Internet Information Server 5.0 with .NET framework • Microsoft SQL server 2000 • ADO.NET • Visual Studio.NET and Dreamweaver MX

4.3 Tools and Technology to be used

Review has been done in purpose of choosing the most suitable tools and technology for this project – InspectionManager. After all the consideration, the proposed tools and technology are as shown in the table below:

Table 4-4: *Tools and Technology Proposed*

Development Model	Rational Unified Process
System Architecture	Three-tier Client-Server Architecture
Application Platform	Windows XP Professional
Web Server	Internet Information Server 5.1 with .NET framework
Programming Language	ASP.NET
Authoring Tool	Visual Studio.NET and Dreamweaver MX
Database Management System	Microsoft SQL Server 2000
Data Access Technology	ADO.NET

Selected System Architecture

For this project, InspectionManager is designed to be **Three-tier architecture** as it is more suitable to be applied in the development of the system to perform its business functionality.

Advantages of Three-tier architecture

- processing can be centralized in the middle-tier
- easier to organize the implementation
- allow different tiers to be developed in different languages
- enhanced secure
- support hundreds of users, making it more scalable
- provides for more flexible resource allocation
- performance balancing

Selected Application Platform

After reviewing on Windows XP, Unix, RedHat Linux and Macintosh, **Windows XP** is chosen to be the project application platform as it support all the tools and technologies that will be used in this project.

Advantages of Window XP

- provides better integration of Windows 9x and Window NT that did Windows 2000
- uses slightly more total memory for the OS to add features than does Windows 2000
- offers significant GUI enhancements
- built-in support for compresses files
- advanced file sorting options
- more stable and improve troubleshooting tools

Selected Web Server

Microsoft Internet Information Server (IIS) is chosen as the system web server, because it comes free with Microsoft Windows XP and includes a set of program for building and administrating web site. Moreover, it is one of the best web server in the market.

Advantages of Microsoft Information Server (IIS)

- support Microsoft ASP.NET technology and web based application that access database

- Easy to install and uninstall
- indexing, performance and security enhancements
- well integrated server administration tools
- easy to configure
- Support window based web authoring and development tool

Selected Programming Language

For this project, **ASP.NET** is chosen as it is the next generation of Microsoft's Active Server Page (ASP), a feature of Microsoft's Internet Information Server (IIS). It allow dynamic web page to be developed by inserting queries to a relational database in the web page.

Advantages of ASP.NET

- Complete Compatibility
- Enhanced Performance
- Power and Flexibility
- World-Class Tool Support
- Simplicity and Manageability
- Rich set of Class Library
- Scalability and Availability
- Security

Selected Authoring Tool

Macromedia Dreamweaver MX and **Microsoft Visual Studio .NET 2003** are both selected as the system authoring tool. However, Dreamweaver MX is used as the primary tool with Visual Studio .NET as secondary tool. Dreamweaver MX supports ASP.NET, programming language that chosen for this project. In Dreamweaver MX, the languages for ASP.NET can be set and the web form can be created by just drag and drop the icons to the specific position.

Advantages of Macromedia Dreamweaver MX

- Enhance productivity using the new integrated workspace, which is shared with Macromedia Flash MX and Fireworks® MX.
- Use one integrated development environment to develop HTML, XHTML, XML, ASP, ASP.NET, JSP, PHP, and Macromedia ColdFusion® websites
- Multiple technology development environment
- Standard and Accessibility support
- Integration with other technologies

Selected Database Management System

Microsoft SQL Server 2000 is the suitable choice for the development of InspectionManager as it works well with databases of any size. Additionally, Microsoft SQL Server 2000 is the most robust database for the windows family.

Advantages of Microsoft SQL Server 2000

- Able to support large-scale database

- User-friendly queries
- High scalability, availability and reliable
- Ease of installation, deployment and use
- Work well with other Microsoft's component
- Can be queried and updated via Web browsers through integration with IIS

Selected Data Access Technology

Reviews have been done in order to choose a suitable data access technology. As a conclusion, **ADO.NET** is chosen as it was designed specifically for the web with scalability, statelessness, and XML in mind. ADO.NET is an evolutionary improvement to ActiveX Data Objects (ADO) that directly addresses user requirements for developing scalable applications.

Advantages of ADO.NET

- Better performance (faster) than ADO
- Complete re-engineering
- Provides data access services in the Microsoft .NET platform using data provider
- Provides data command and data readers to communicate directly with data source
- XML support and integration
- Disconnect Access Model

Chapter 5 System Design

System design is considered as an important part of the system development process. It sits at the technical kernel of the system development process. System design is utilized and applied regardless of what kind of development model or standard being used. In shorts, system design is a process to convert the conceptual ideas from requirement specification in system analysis into more technical specification.

In system design phase, the system requirements gathered during the analysis phase are transmitted into a representation or a "blueprint" for constructing the system. Initially, this representation depicts a holistic view of system. There are series of activities involved in system design process, namely, analyzing, coding, and testing the system or prototyping to ensure that it conform to the software specifications and requirements, which have been defined in earlier stages.

Under this chapter, the system design will be discuss in the following categories:

- System Architecture
- System Functionality Design
- Database Design
- Interface Design

5.1 System Architecture

A system's architecture is the framework that describes how system components interact and work together to achieve total system goals.

5.1.1 Client-Server Computing

In order to use services available on an internet or intranet network, application programs that running at two end computers and communication with each other are needed. In other words, the application programs are the entities in an internet or intranet network that communicate with each other, not the computers and users.

The application programs follow the Client-server model strategies. This model is based on the distribution of functions between two types of independent and autonomous processes: client and server. A client is any process requests specific server for server processes while a server is a process that provides requested services for clients.

Client-Server model strategies:

- As shown in Figure 5.1, an application program called the **client**, running on the local machine, request a service from another application program - the **server** that running on the remote machine.

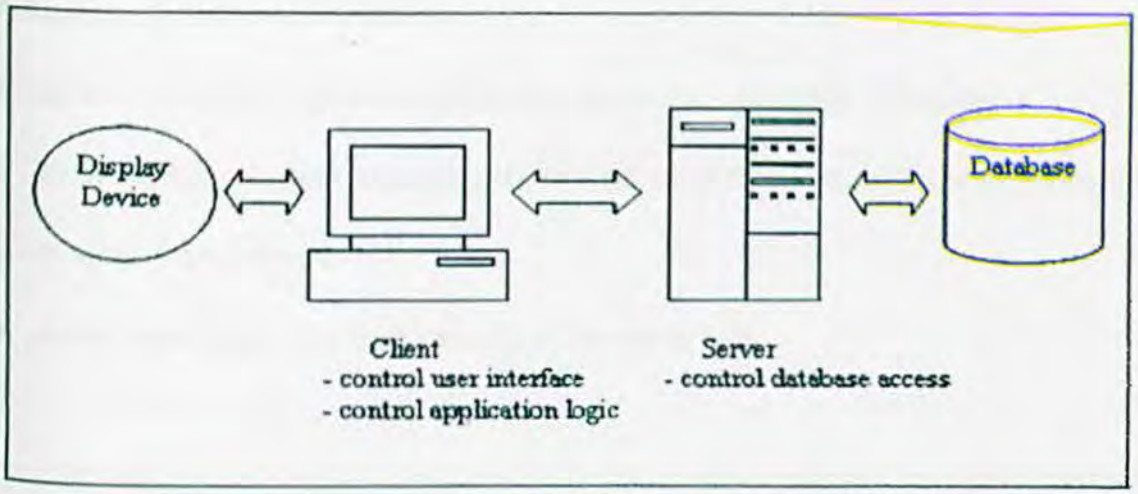


Figure 5-1: Client-Server Connection

- A server can be provided to many clients, not just a particular client. This means that the client-server relationship is many-to-one where many client can use the services of one server
- A client program, which requests a service, should run only when it is needed while the server program, which provides service, should be running all the time waiting for incoming request.

Refer Chapter 2: System Architecture for advantages of client-server computing.

5.1.2 Three-tier Architecture

There are generally two kinds of client-server architecture to choose from: two-tier and three-tier. The choice should be made based on the scope and complexity of a project, the time available, and the expected enhancement of the system.

Advantages of three-tier architecture:

- an increasing number of homogeneous product are available off the shelf
- there are little network bandwidth to be used since the information is to be displayed is sent on the network
- some upgrades can be done entirely at the server side

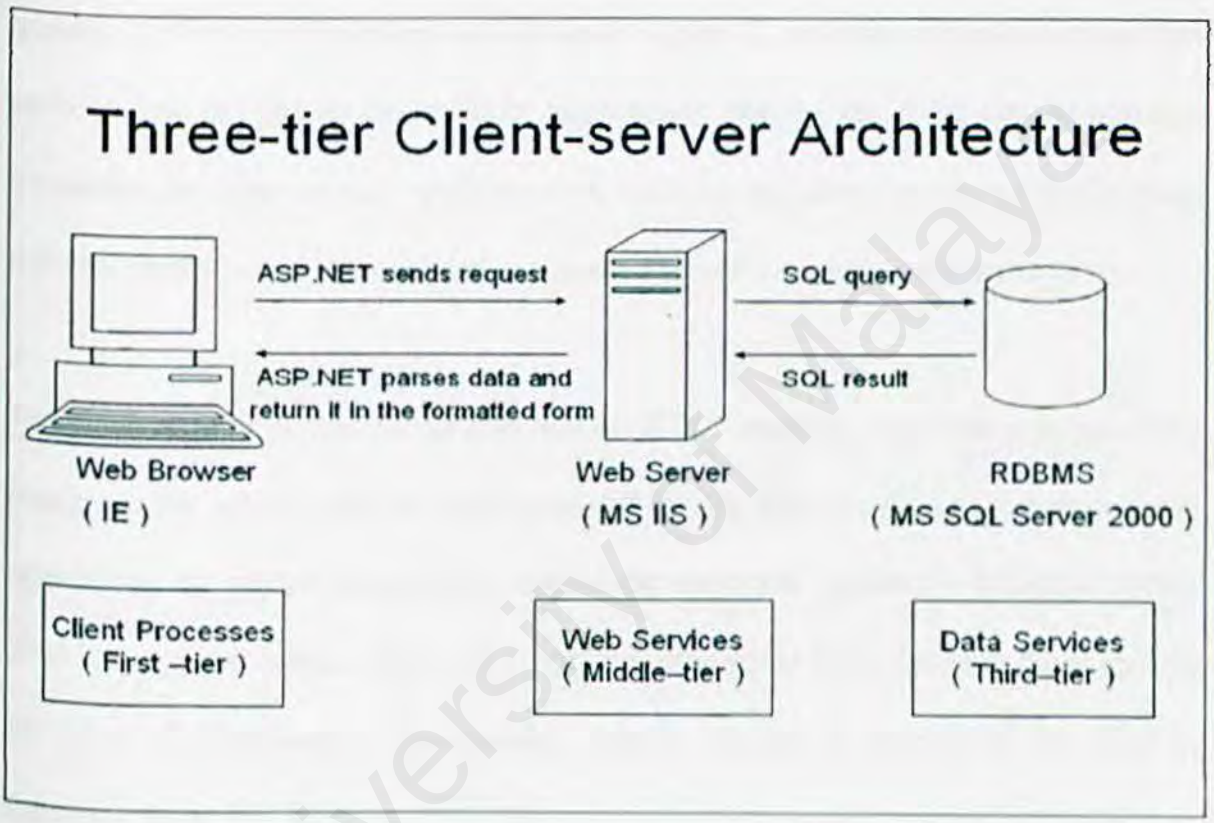


Figure 5-1: System Architecture Design

The three-tier client-server architecture builds on the traditional two-tier approach. It attempts to overcome some of the limitation of the two-tier scheme. The first tier refers to the client, the second tier or middle tier is the business logic and server, while the third tier consists of the associated database. From a software perspective, the three tier

are client processes (Web browser, in this case – Internet Explorer), web services (Business logic and web server, MS IIS is chosen) and data services (RDBMS – MS SQL Server 2000 is selected). Interaction between client and server operate the same way as they do in the two-tier architecture (refer Chapter 2: System Architecture).

The first tier is where services reside, such as text input. The middle tier can perform queuing, application execution, and database staging. It provides process management services that are shared by multiple applications. As for the third tier, it provides comprehensive data services which include database operations supported by database software, and other services needed to support a robust electronic commerce server.

The client request is first formulated into an HTTP message, and then it is sent over internet to the server – MS IIS and examined by this MS IIS. Analysis of the request reveals that the request requires the help of the relational database – MS SQL Server 2000 to query and retrieve data. After that, the information from database flows back to the server and followed by data parsing. Finally, the data is returned to the client in formatted form.

5.2 System Functionality Design

System functionality design based on the system requirements stated in Chapter 4. It translates the system requirement into system functionality. System functionality design explains how the modules interact with one another and also the functionalities in each module.

5.2.1 Sequence Diagram

A sequence diagram is known as an art interaction diagram as it allows user to graphically display various interactions among objects. It shows a set of objects and also the message sent and received by those objects. Sequence diagrams establish the roles of objects and help provide essential information to determine class responsibilities and interfaces. This sequence diagram is used to specify and implement the control aspects of a system. The significant advantages of sequence diagram are it clearly depict the sequence of events, show when objects are created and destroyed, are excellent at depicting concurrent operations, and are invaluable for hunting down race conditions.

Table 5-1: Symbol and Description Used in Sequence Diagram

Elements/Action	Symbol	Description
Actor		represents a role played in relation to the business by someone or something in the business environment
Boundary Class		represents an interface between the system and some entity outside the system: a person or another system

Entity		used to model information and associated behavior that must be stored
Call		A call action invokes an operation on an object
Return		A return action is the return of a value to the caller, in response to a call action

Below are the sequence diagrams of InspectionManager.

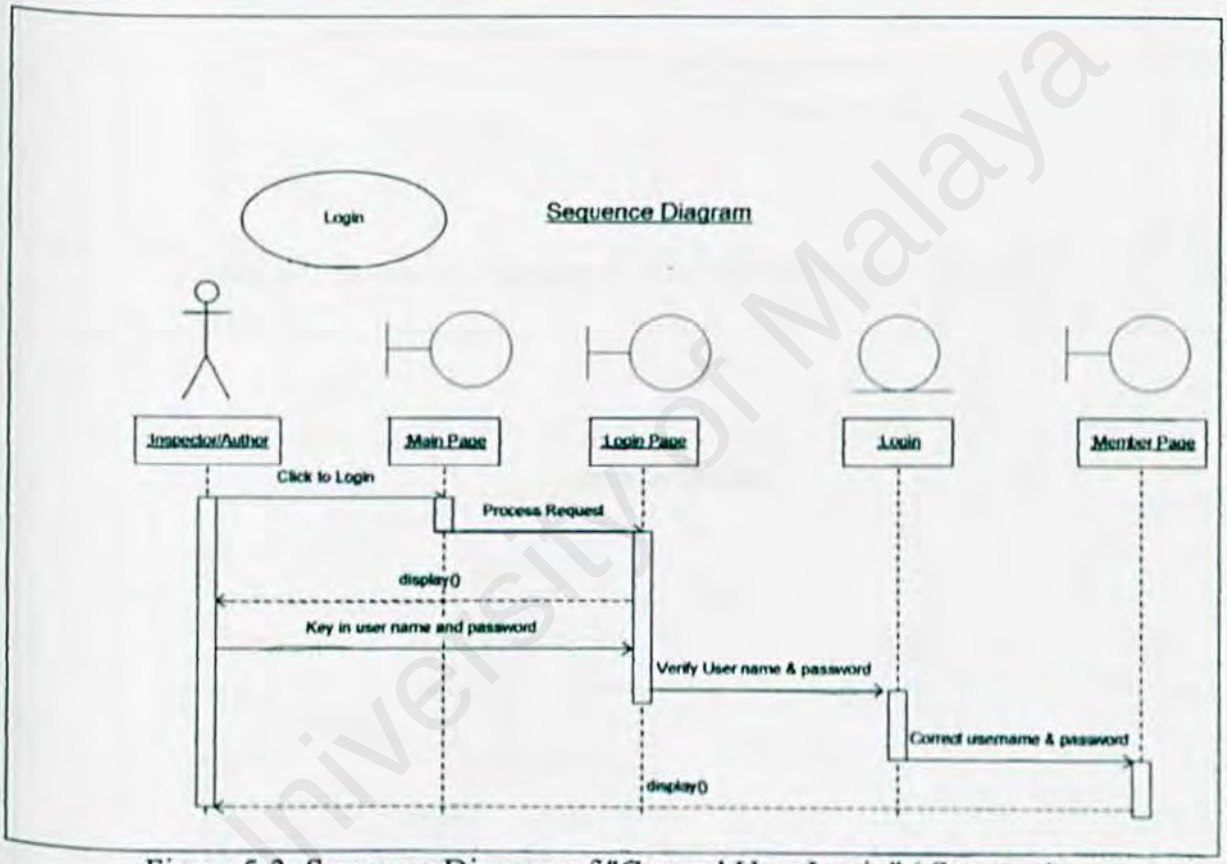


Figure 5-3: Sequence Diagram of "General User Login" (Success)

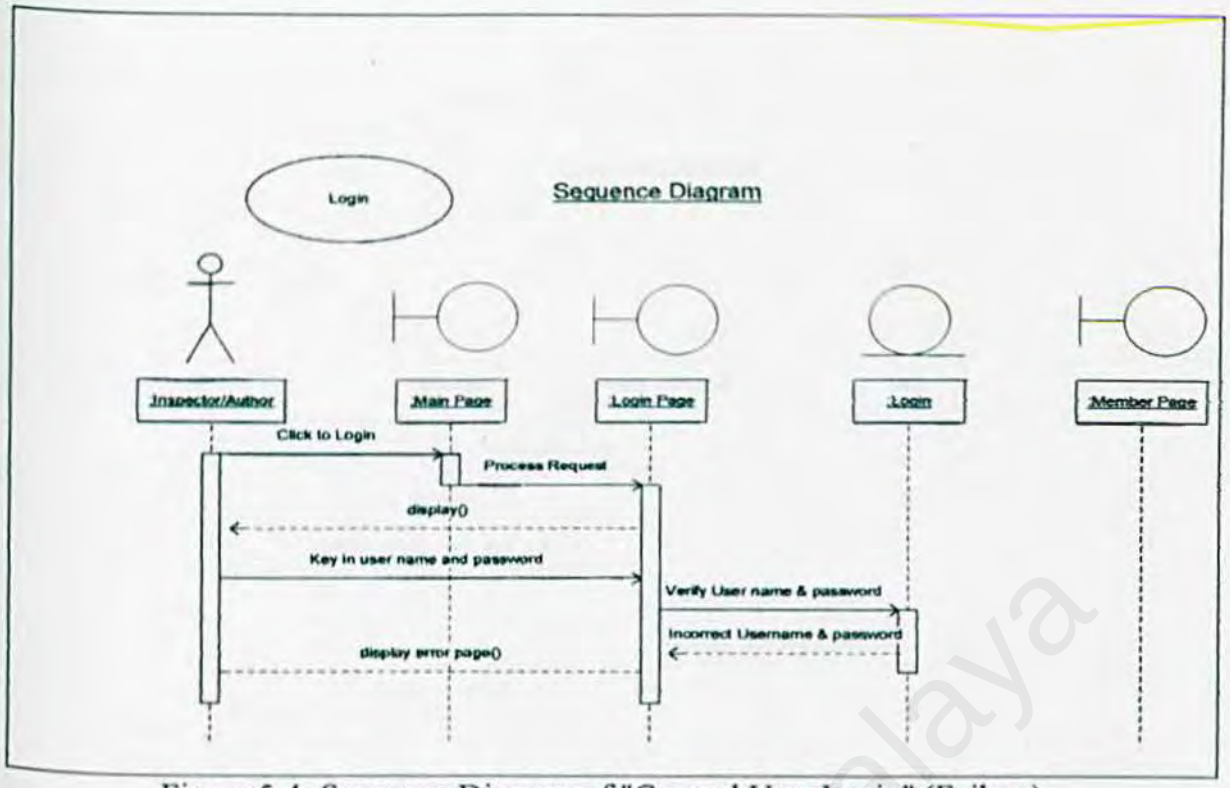


Figure 5-4: Sequence Diagram of "General User Login" (Failure)

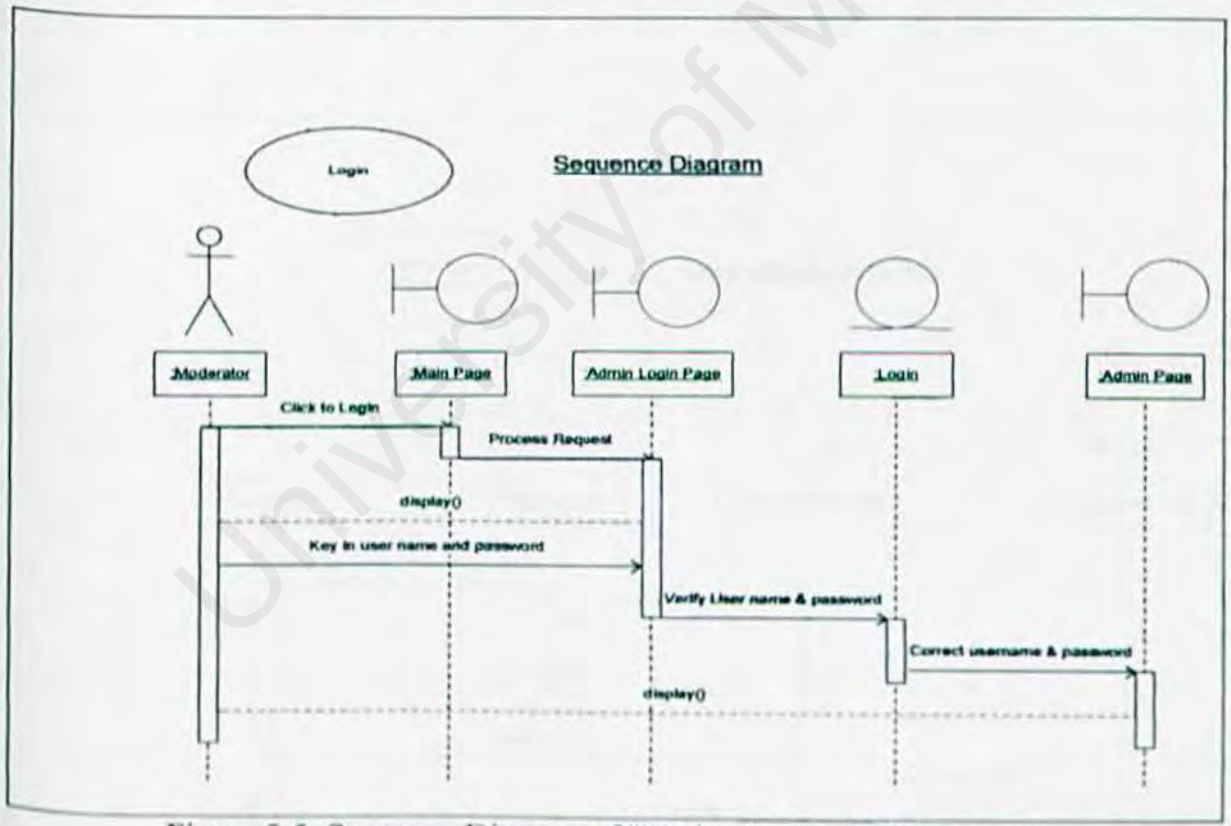


Figure 5-5: Sequence Diagram of "Moderator Login" (Success)

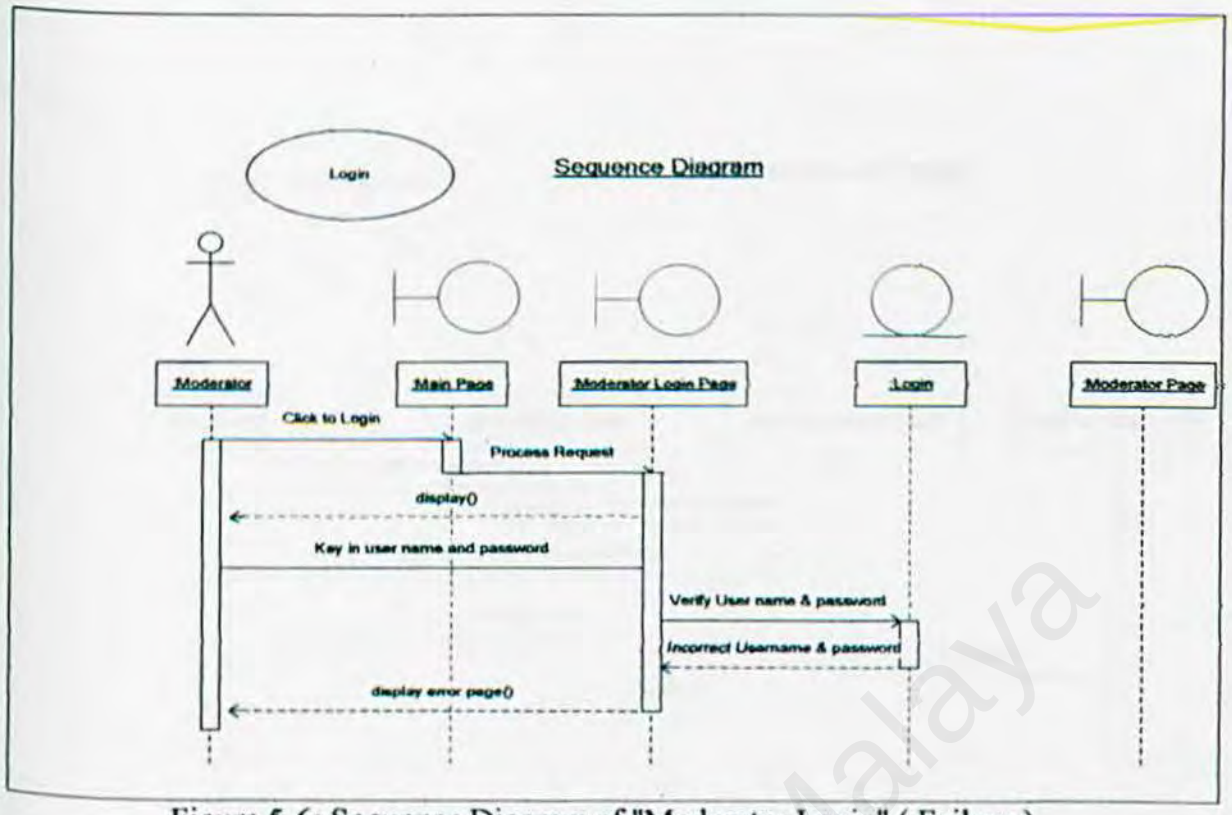


Figure 5-6: Sequence Diagram of "Moderator Login" (Failure)

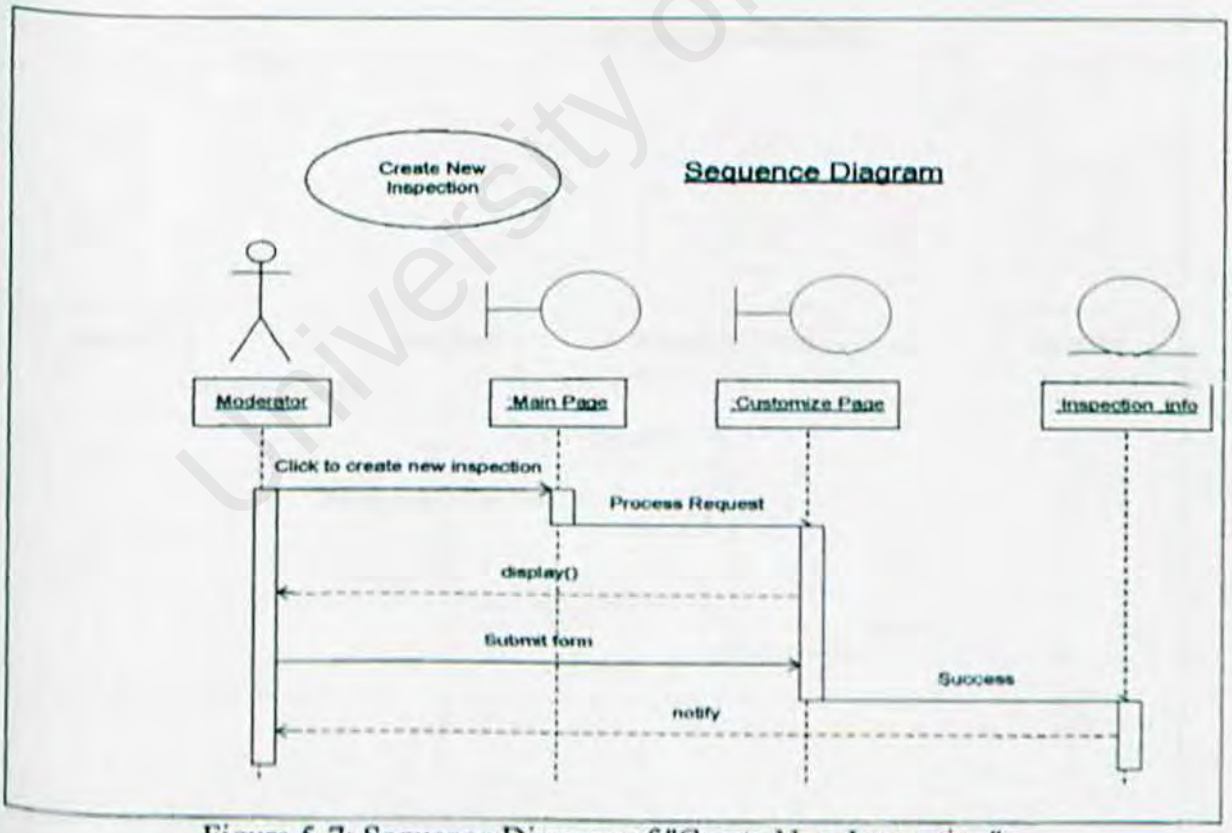


Figure 5-7: Sequence Diagram of "Create New Inspection"

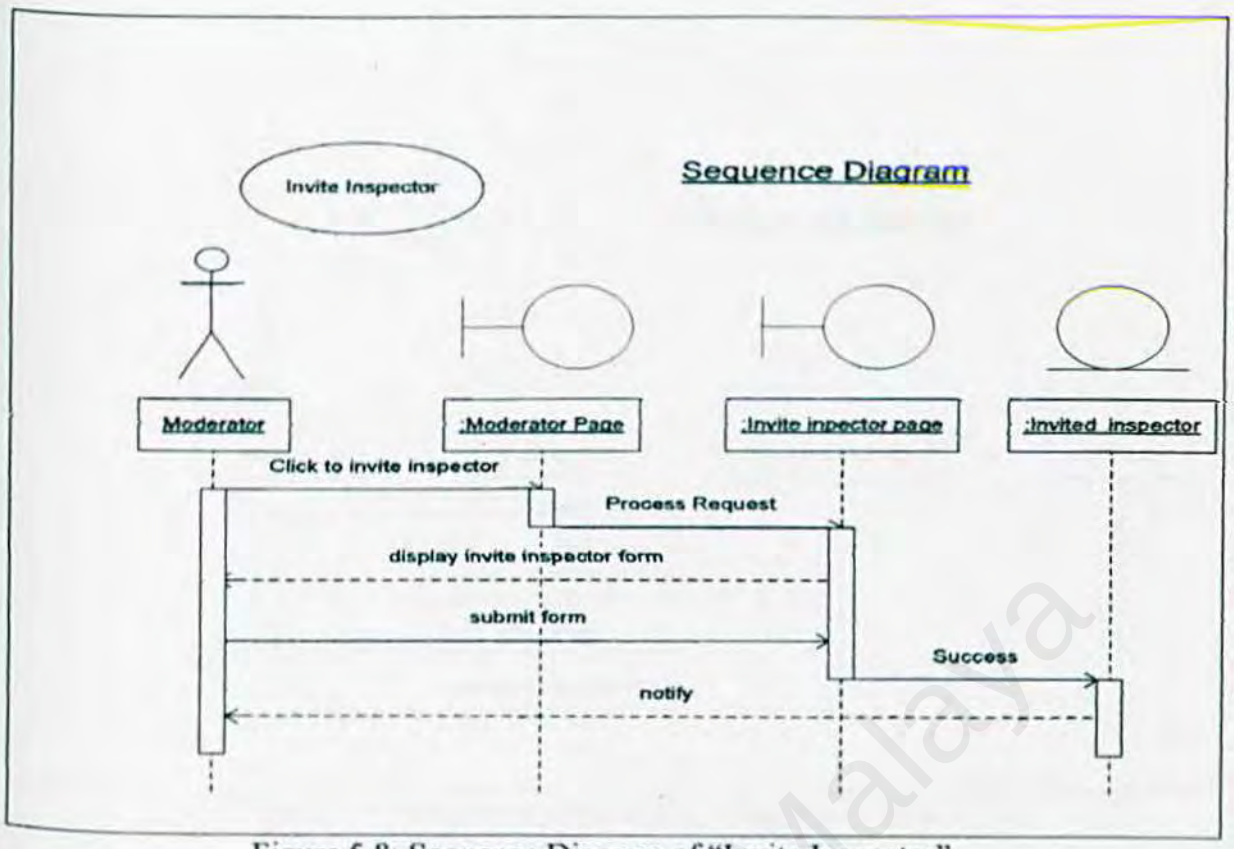


Figure 5.8: Sequence Diagram of "Invite Inspector"

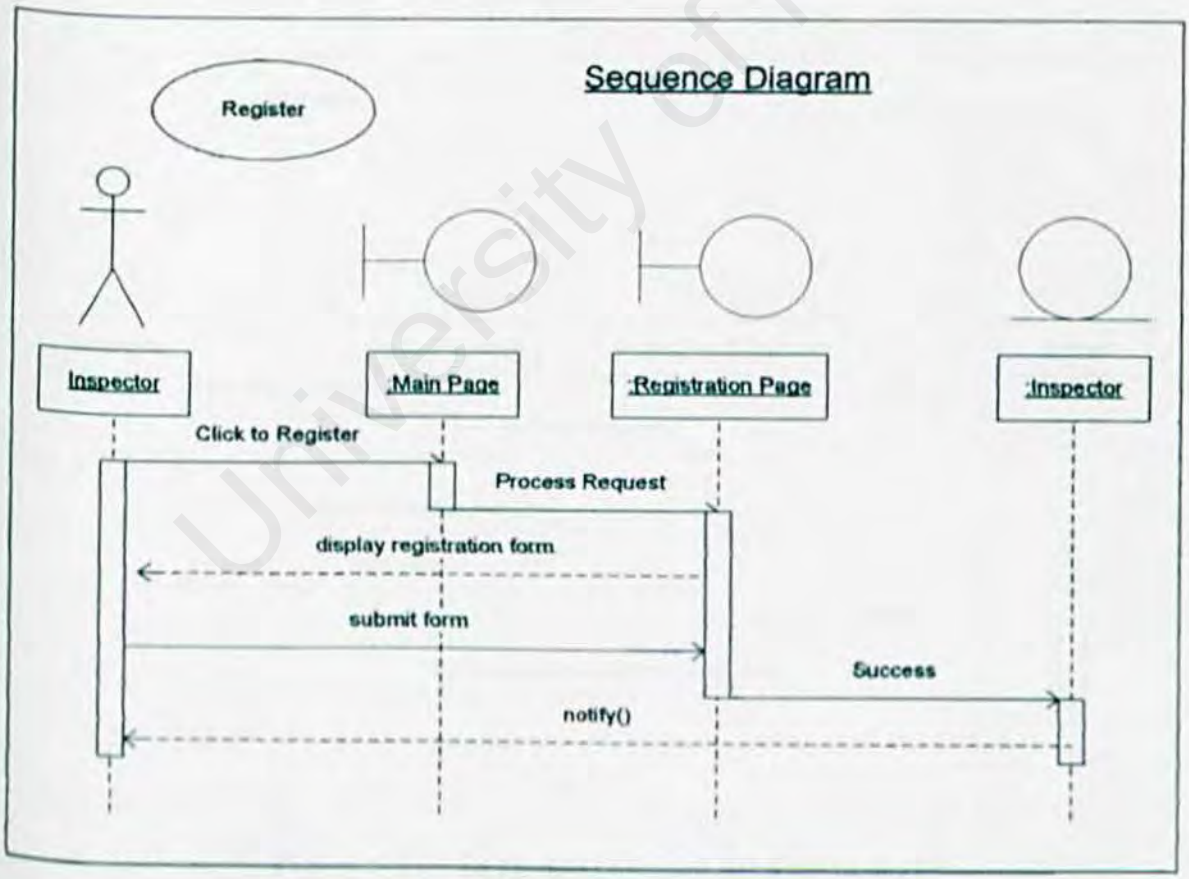


Figure 5.9: Sequence Diagram of "Register"(Success)

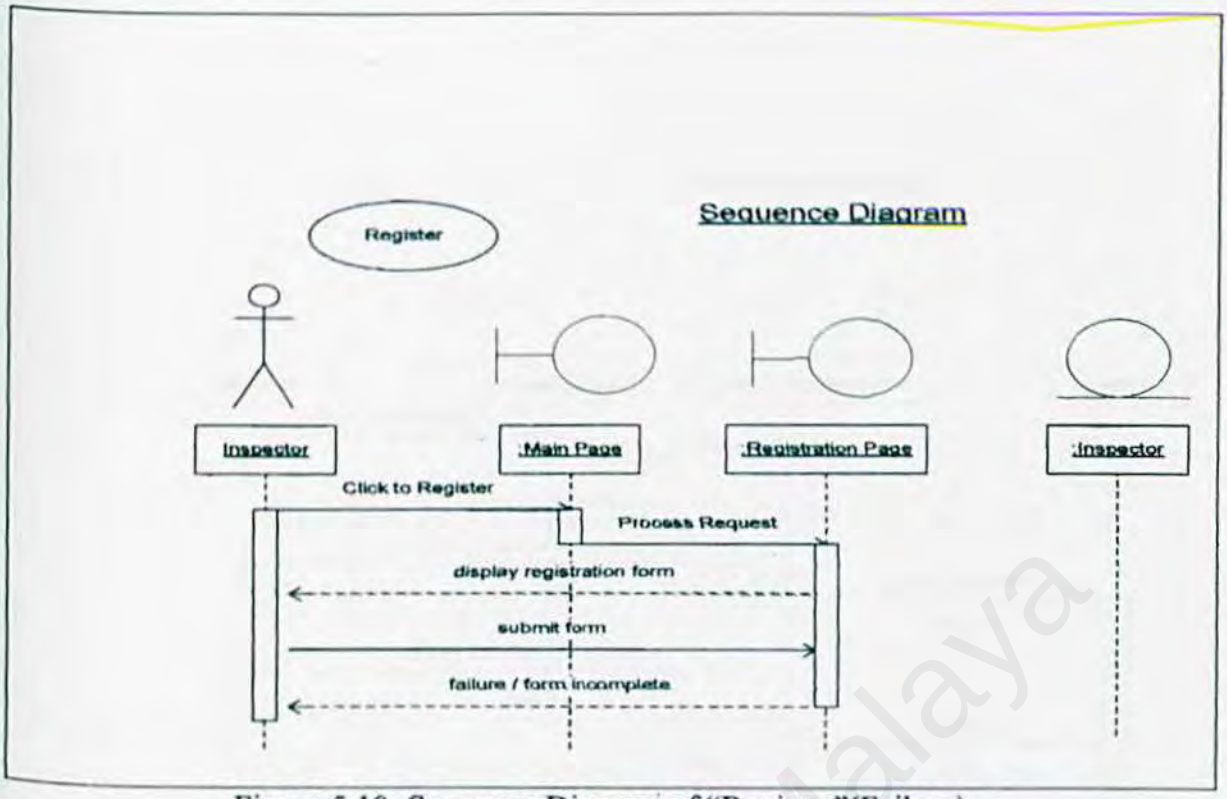


Figure 5.10: Sequence Diagram of "Register"(Failure)

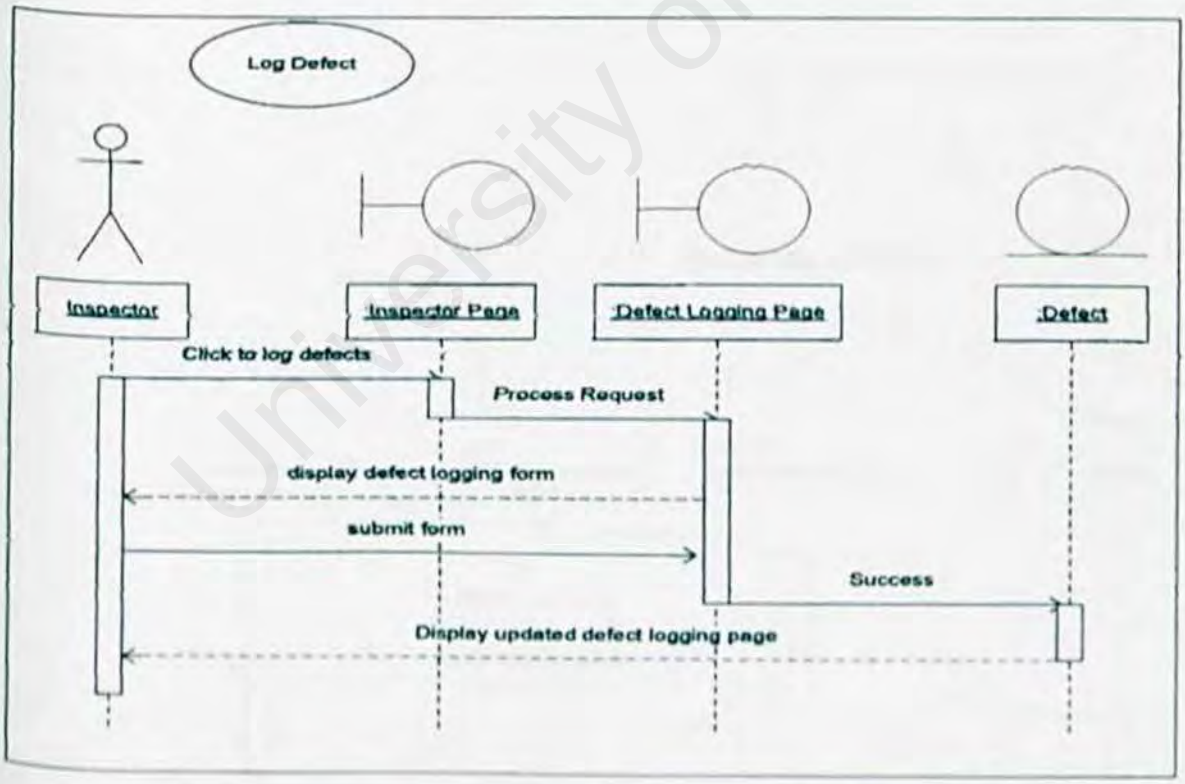


Figure 5.11 : Sequence Diagram of "Log Defect"

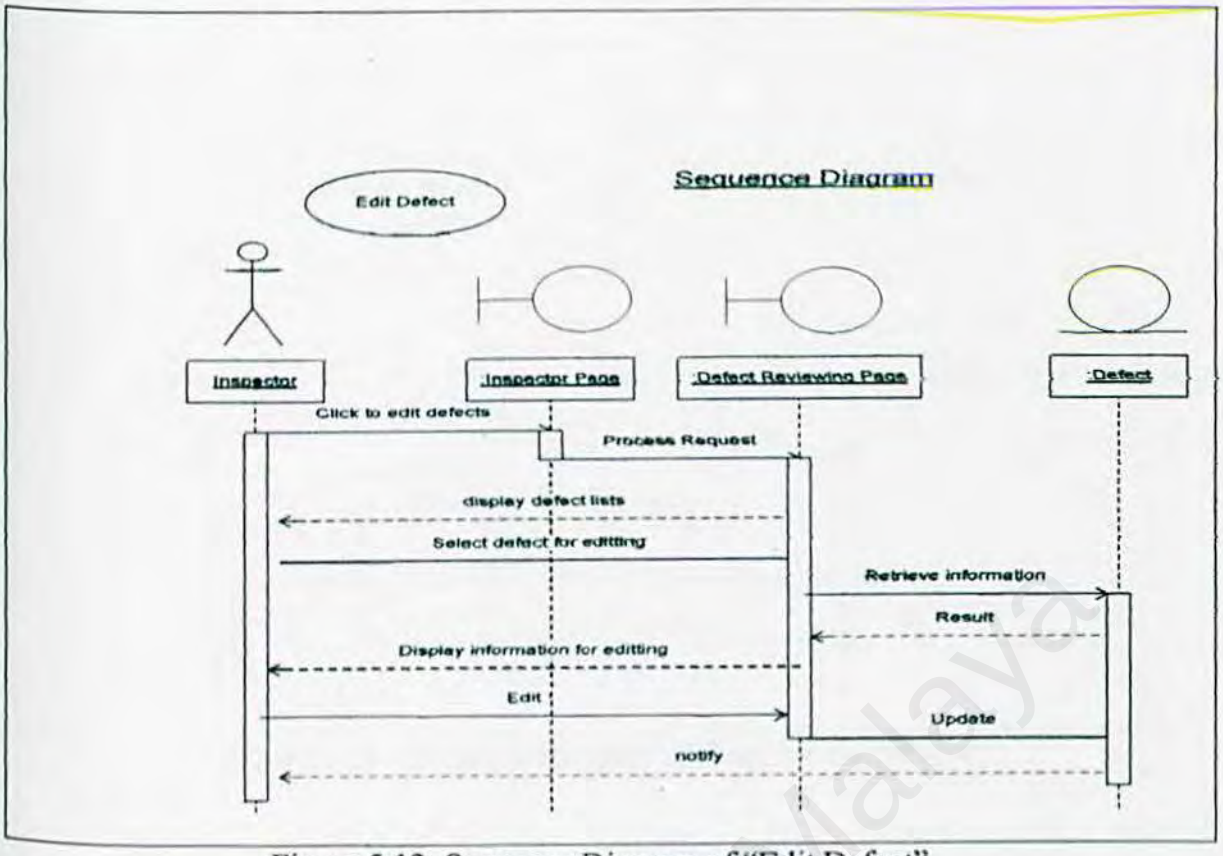


Figure 5.12: Sequence Diagram of "Edit Defect"

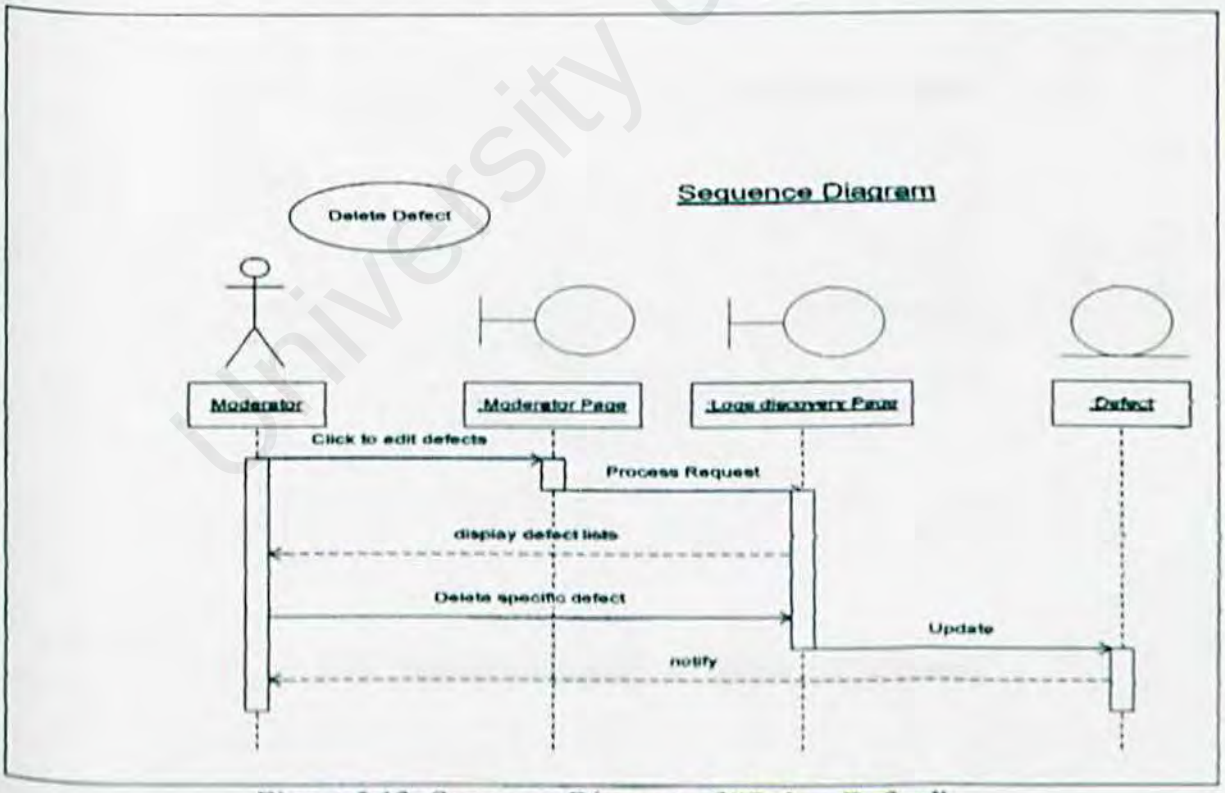


Figure 5.13: Sequence Diagram of "Delete Defect"

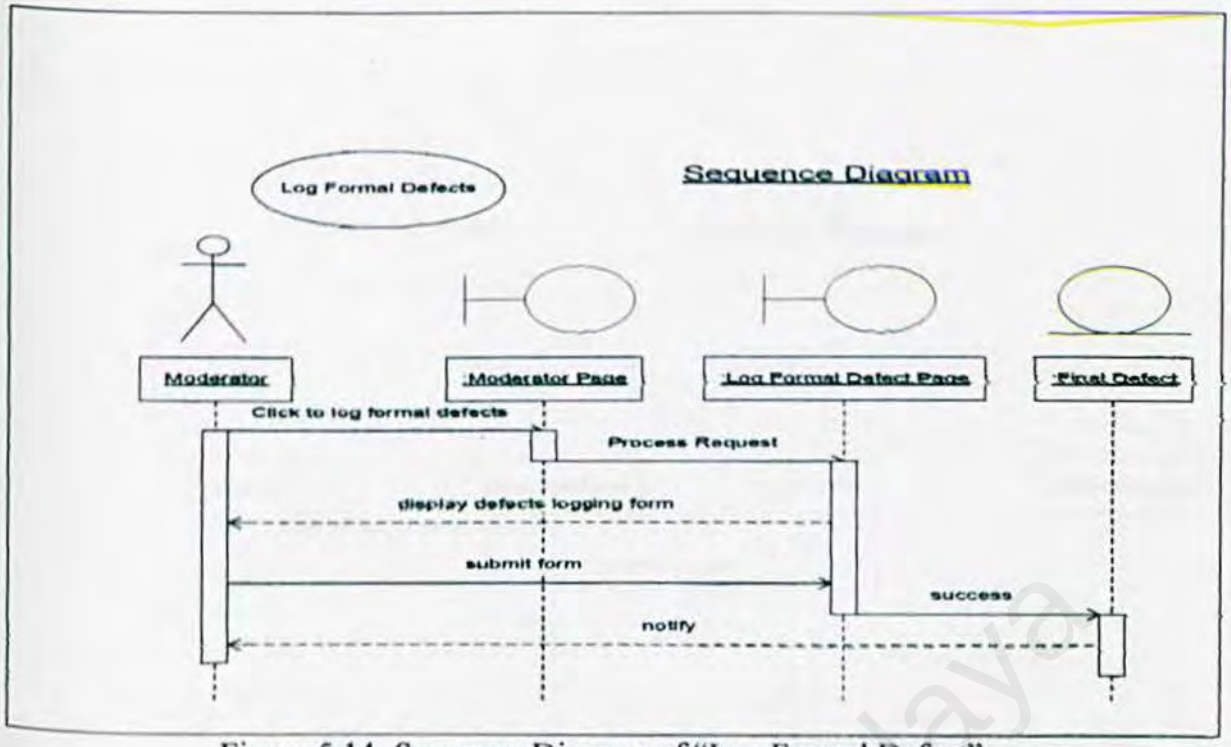


Figure 5.14: Sequence Diagram of "Log Formal Defect"

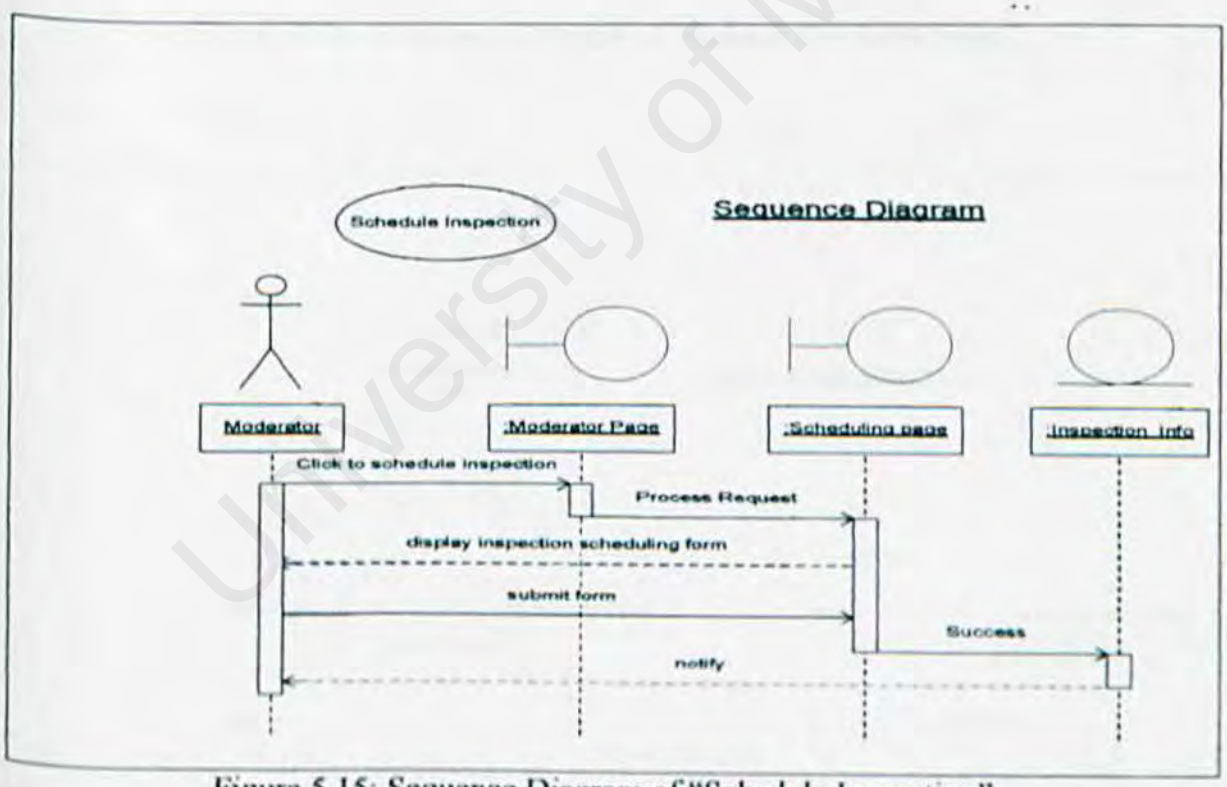


Figure 5.15: Sequence Diagram of "Schedule Inspection"

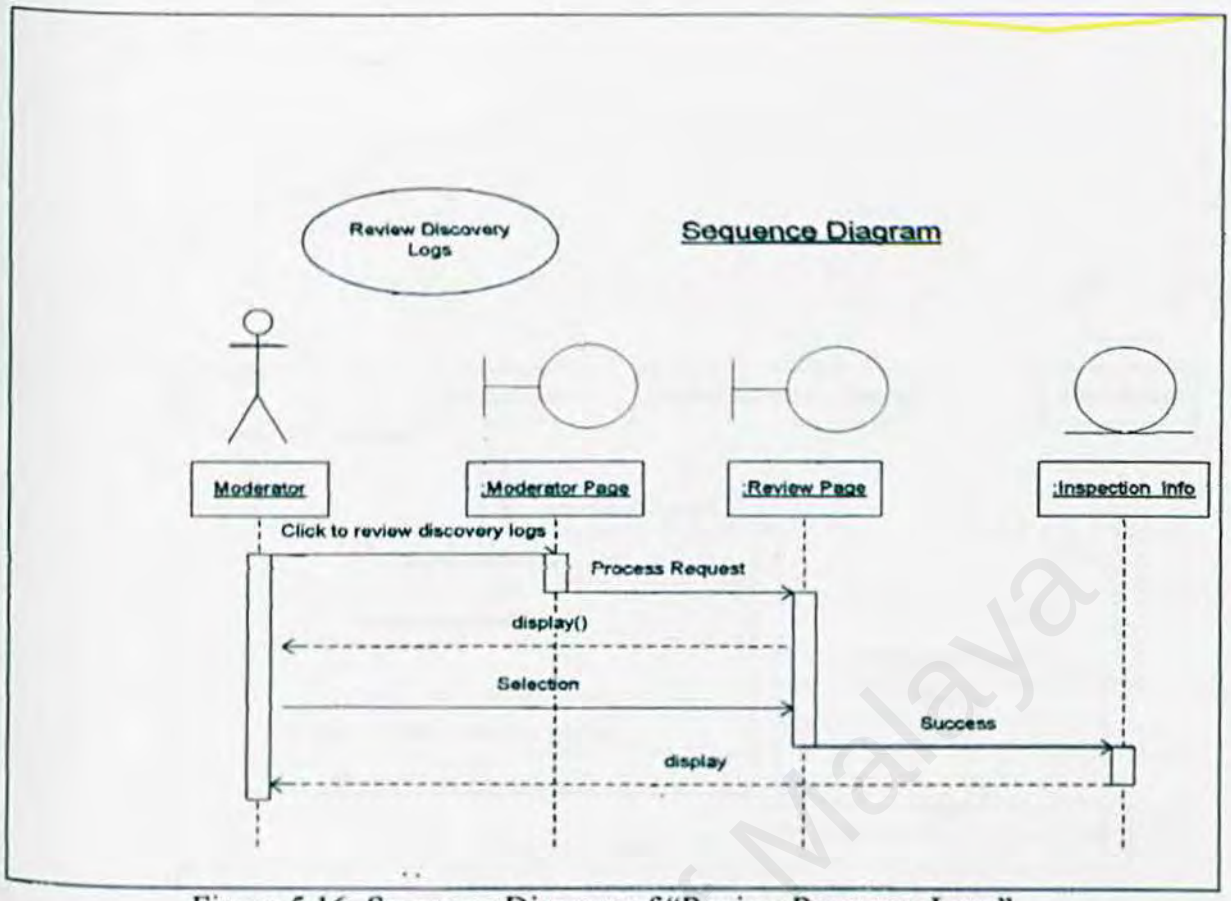


Figure 5.16: Sequence Diagram of "Review Recovery Logs"

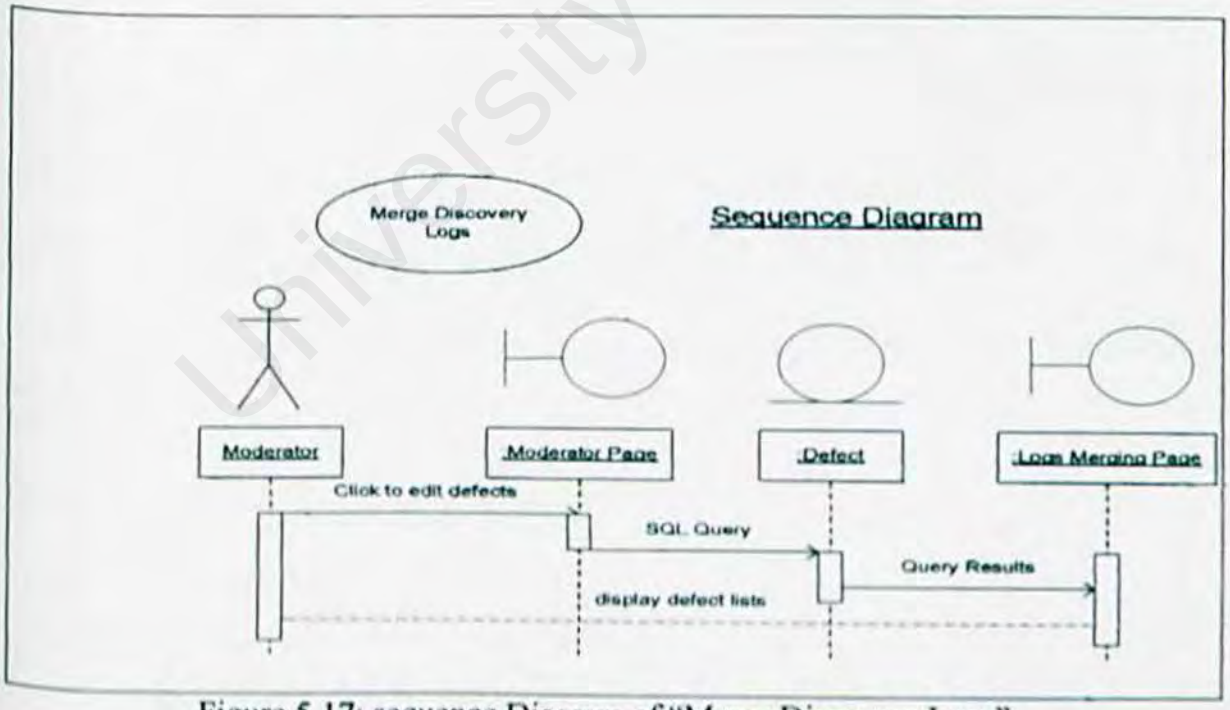


Figure 5.17: sequence Diagram of "Merge Discovery Logs"

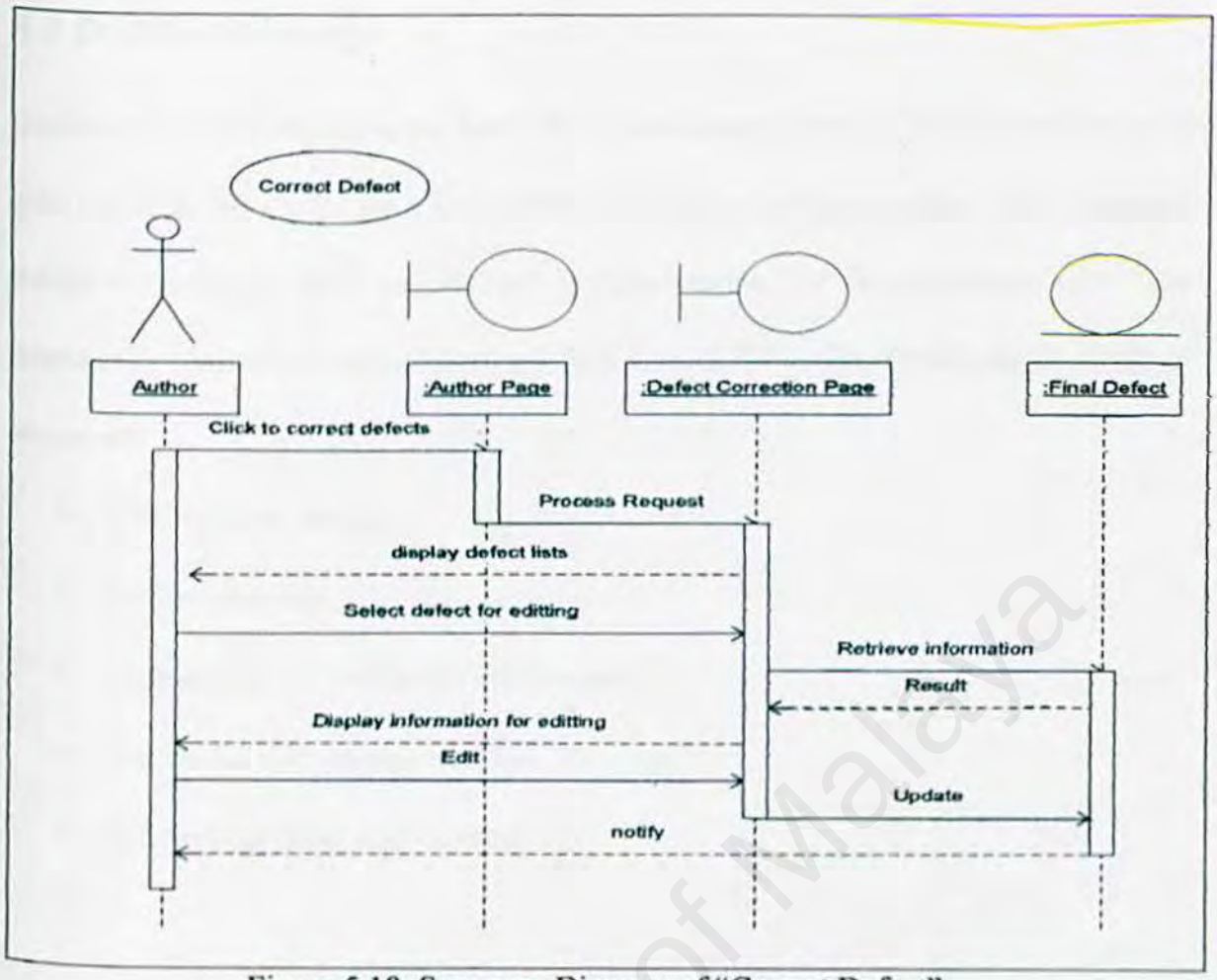


Figure 5.18: Sequence Diagram of "Correct Defect"

5.3 Database Design

Data storage is considered as the heart of an information system. It is a central source of data meant to be shared by many users for a variety of applications. The relational database model is used in database implementation for InspectionManager. The database is constructed using Microsoft SQL Server 2000. The objectives of database design are:

- Efficient data storage
- Data availability
- Data accuracy, consistency and integrity
- Purposeful information retrieval
- Efficient updating and retrieval

5.3.1 Class Diagram

A class diagram shows the existence of classes, interface, collaborations and their relationships in the logical design of a system. A class diagram may represent all or part of the class structure of a system that is being developed. It is used to model the static design view of a system. The relationships among the classes in the class diagram provide the stepping stone for the structure of the system. In shorts, class diagrams are used for a wide variety of purposes, including both conceptual/domain modeling and detailed design modeling.

A class icon is drawn as a 3-part box, class name as the top part, a list of attributes (with optional types and values) in the middle part, and a list of operations (with optional argument lists and return types) is the bottom part.

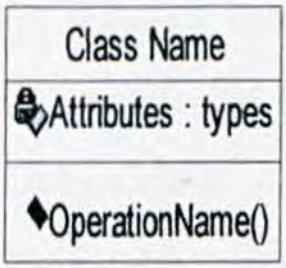


Figure 5-19: Class Notation

Below is the Class Diagram for InspectionManager

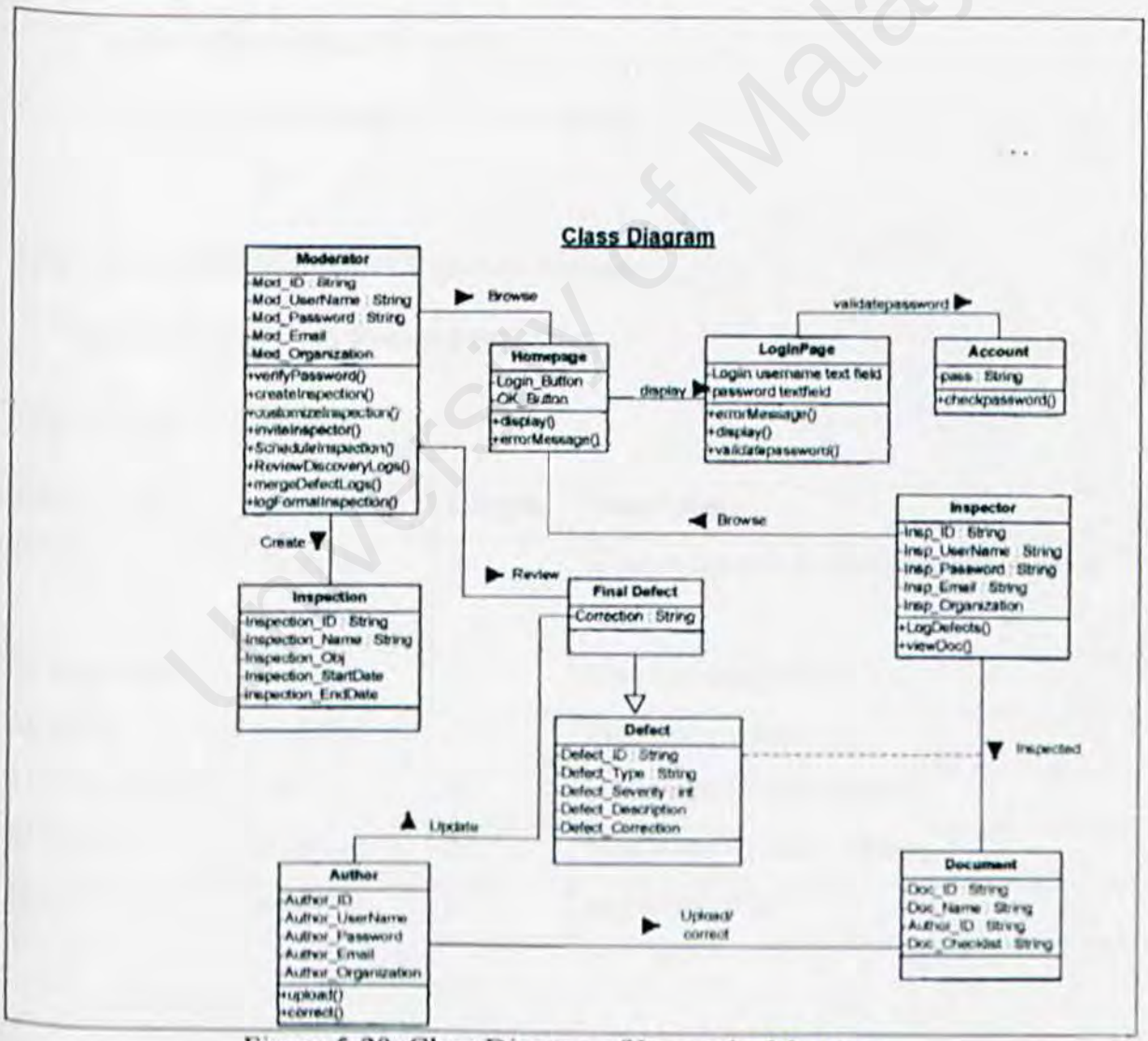


Figure 5-20: Class Diagram of InspectionManager

5.3.2 Data Dictionary

Data dictionary (DD) or metadata can be defined as descriptions of the database structure and contents where it defines the field, field type and description of each table. DD is a repository of all elements in the system. It is a logical characteristic of current system data stores. The importance of data dictionary:

- Manage details in large system
- Document the features of the system
- Communicate a common meaning for all system elements
- Facilitate analysis of the details (to evaluate characteristic and determine where system change should be made)
- Locate errors and omissions in the system

Below is the data dictionary of InspectionManager

***: primary key #: foreign key**

Table 5-2: Table of Moderator

Field Name	Data Type	Length	Description
*M_ID	nvarchar	8	Unique number generated for Moderator created
M_RegisDate	datetime	8	Date for registration
M_Name	nvarchar	50	Moderator's name
M_Organization	char	10	Moderator's Organization
M_Email	nvarchar	50	Moderator's e-mail address
Insp_ID #	nvarchar	8	Inspection's ID

Table 5-3: Table of Inspector

Field Name	Data Type	Length	Description
*I_ID	nvarchar	8	Unique number generated for Inspector created
I_RegisDate	datetime	8	Date for registration
I_Name	nvarchar	50	Inspector's name
I_Organization	char	10	Inspector's Organization
I_Email	nvarchar	50	Inspector's e-mail address
Insp_ID #	nvarchar	8	Inspection's ID

Table 5-4: Table of Author

Field Name	Data Type	Length	Description
*A_ID	nvarchar	8	Unique number generated for Moderator created
A_RegisDate	datetime	8	Date for registration
A_Name	nvarchar	50	Inspector's name
A_Organization	char	10	Inspector's Organization
A_Email	nvarchar	50	Inspector's e-mail address
Insp_ID#	nvarchar	8	Inspection's ID

Table 5-5: Table of Inspection

Field Name	Data Type	Length	Description
*Insp_ID	nvarchar	8	Unique number generated for Moderator created
Insp_StartDate	datetime	8	Date for inspection registration
Insp_EndDate	datetime	8	Date for inspection ended
M_ID#	nvarchar	8	Unique number generated for Moderator created

Insp_Objective	nvarchar	500	Inspection objective
D_ID#	nvarchar	8	Unique number generated for Document created

Table 5-6: Table of Document

Field Name	Data Type	Length	Description
*Doc_ID	nvarchar	8	Unique number generated for document created
Doc_Name	nvarchar	20	Document's name
A_ID#	nvarchar	20	Document's Author ID
Insp_ID#	nvarchar	20	Unique number generated for Inspection created

Table 5-7: Table of Login

Field Name	Data Type	Length	Description
*Login_ID	nvarchar	8	Login user ID
Login_UserName	nvarchar	20	Admin login name
Login_Password	nvarchar	20	Admin password
Login_Status	nvarchar	20	User position—Moderator, Author or Inspector

Table 5-8: Table of Defect

Field Name	Data Type	Length	Description
*Def_ID	nvarchar	8	Unique number generated for Defect created
Def_Type	nvarchar	20	Defect type such as Redundancy, Ambiguous

Def_Severity	nvarchar	20	Defect Severity such as Major, Minor
Def_Description	nvarchar	1000	Defect Description
Def_ReqNum	nvarchar	20	Defect located at which requirement location
Def_PageNum	nvarchar	8	Defect located at which page
I_ID#	nvarchar	8	Inspector's ID
Insp_ID#	nvarchar	8	Inspection_ID

Table 5-9: Table of Formal Defect

Field Name	Data Type	Length	Description
*Formal_ID	nvarchar	8	Unique number generated for Formal Defect created
Def_Type	nvarchar	20	Defect type such as Redundancy, Ambiguous
Def_Severity	nvarchar	20	Defect Severity such as Major, Minor
Def_Description	nvarchar	1000	Defect Description
Def_ReqNum	nvarchar	20	Defect located at which requirement location
Def_PageNum	numeric	8	Defect located at which page
I_ID#	nvarchar	8	Inspector's ID
Insp_ID#	nvarchar	8	Inspection_ID
Corr_Description	nvarchar	1000	Author's description about the defect and correction made

5.4 User Interface Design

The Human Computer Interface (HCI), commonly known as user interface is doorway into an interactive software application. The user interface design describes how software communicates within itself, to system that interoperates with it, and also with human being who use it. An interface is a set of commands through which a user communicates with the library automation system. The user interface is one of the most important parts of any program because it determines how easily user can makes the system do what he or she want.

Nowadays, there are two types of user interface: a command-driven interface (text based) and a menu-driven interface (graphical user). A command-driven interface involves user entering commands while a menu-driven interface is one in which user selects command choices from various menus displayed on screen. Below are several principles that need to be highlighted during the system interface design.

- Consistency
- Recoverability
- Confirmation and verification message
- Responsiveness
- Reverse action

5.4.1 Graphical User Interface

A Graphical User Interface (GUI) is a graphical (rather than purely textual) user interface to the library automation system that intended to provide direct correlation between the visual stimulus on the computer screen and the desired response from the library system. A good GUI should be intuitive, minimum the need for user to memorize things and must be interesting to look at.

The advantages of Graphical User Interface:

- Certainty, that the interface will look the same on any operating system with all mainstream browsers
- Enables frequent users to use shortcuts
- Attractive
- Offer informative feedback
- User friendly
- Easy to use
- Able to communicate the information to the user easily

5.4.2 User Interface for InspectionManager

InspectionManager is a web based system; therefore the user interfaces are designed according to web page style. Typically the user interface is to help users to navigate through web pages and make request.

Below are the InspectionManager's User Interface

i) Main Page

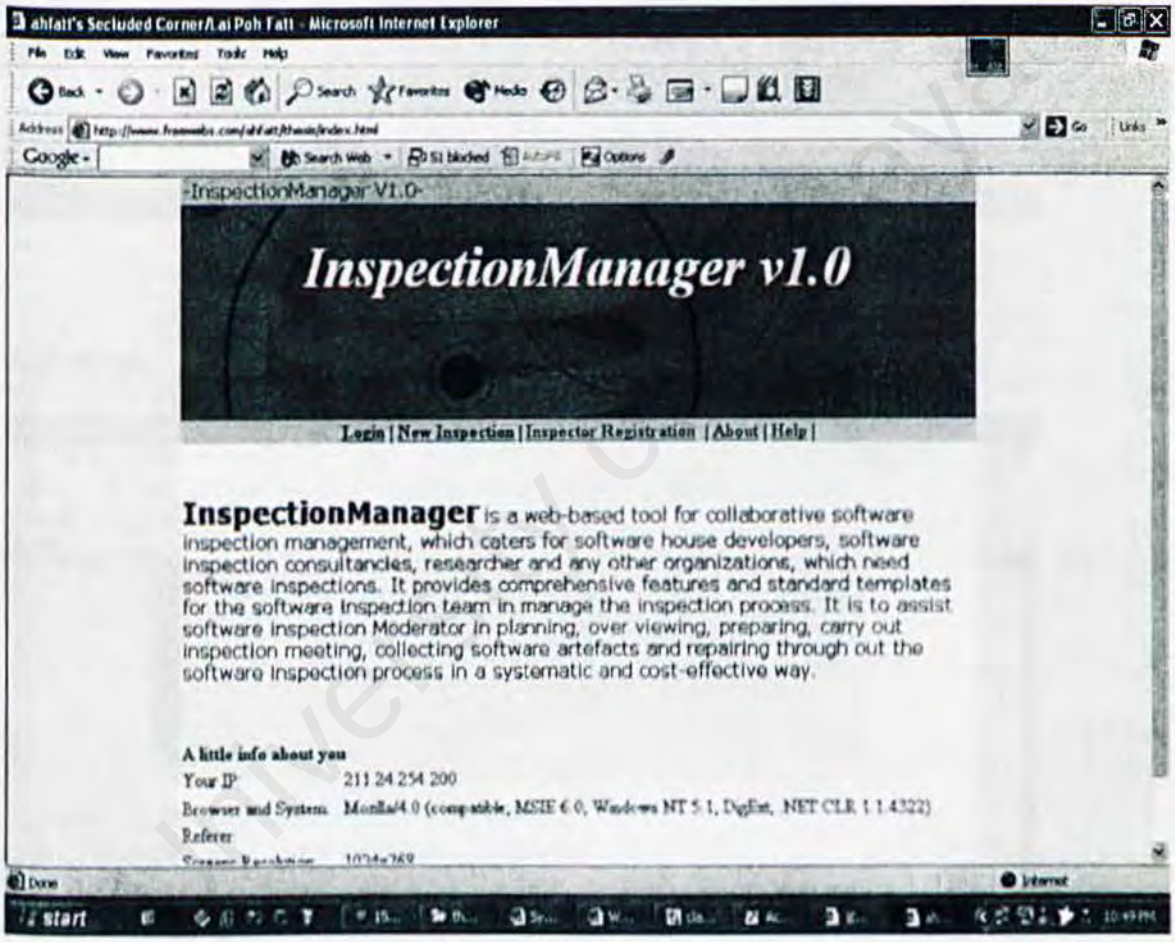


Figure 5-21: Main page of InspectionManager

ii) Login Page

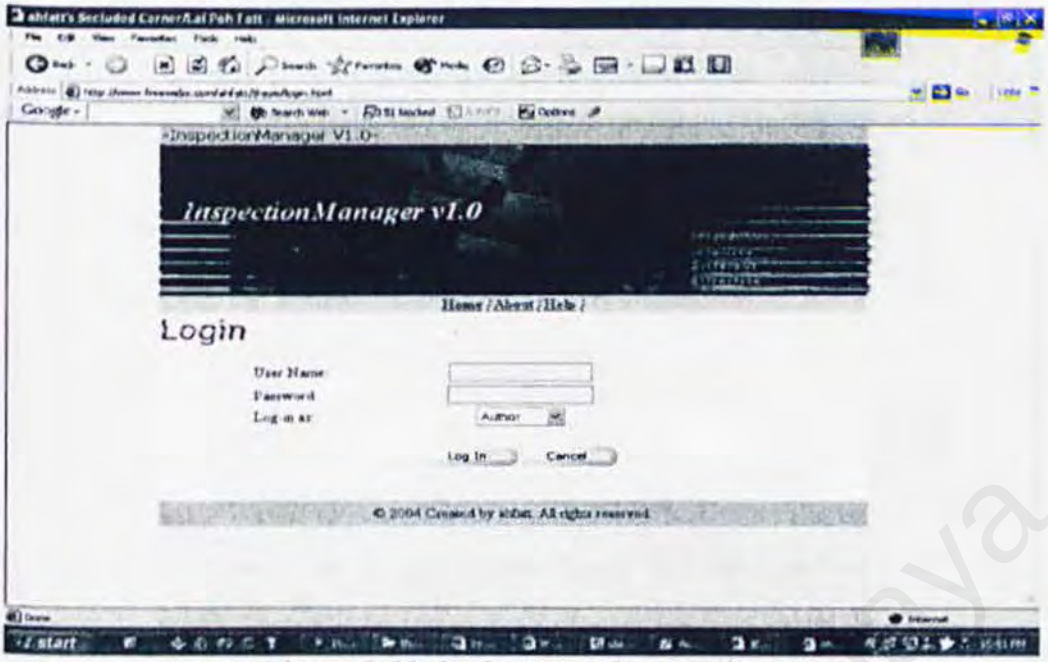


Figure 5-22: login Page of InspectionManager

iii)Register New Inspection

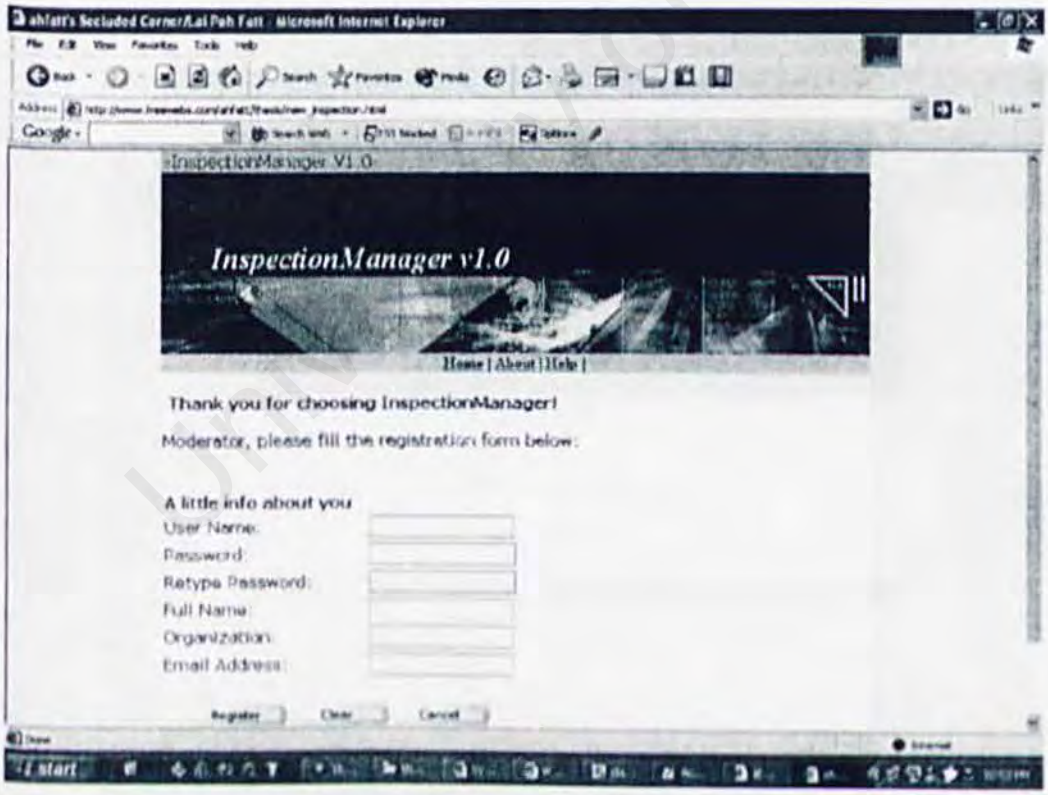


Figure 5-23: Register New Inspection Page

iv) Moderator Page

Figure 5-24: Moderator Page

v) Logout Page

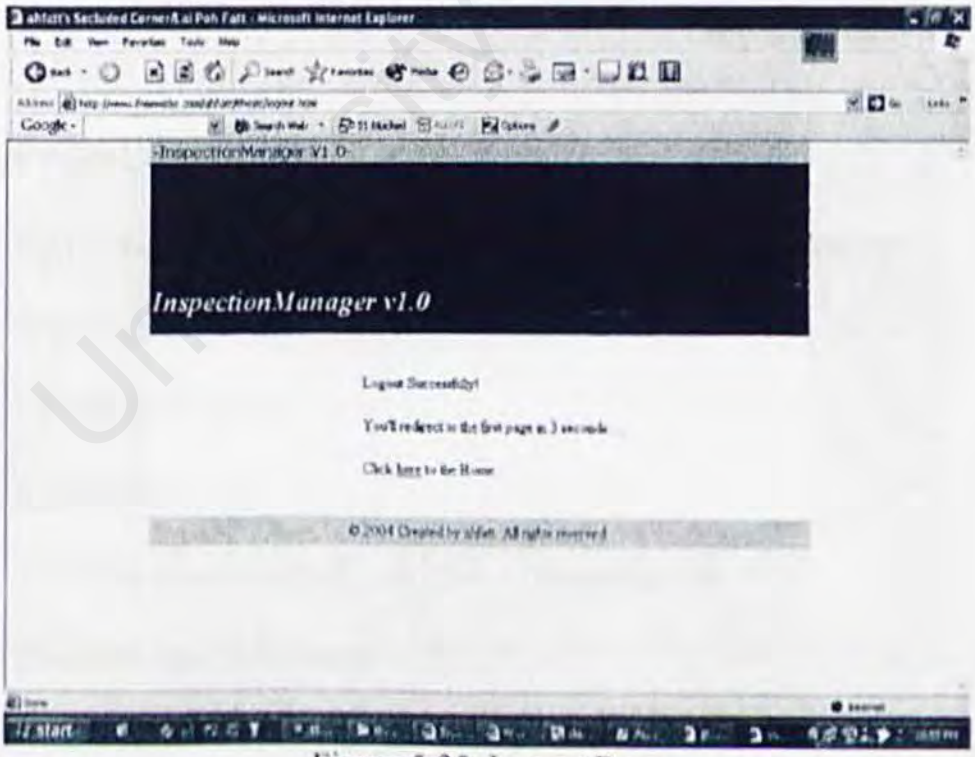


Figure 5-25: Logout Page

Chapter 6 System Implementation

During this phase, the design model of InspectionManager is transformed into workable product. Therefore, system implementation involved the translation of the software representation produces by the design into a computer understandable form. It involves coding of the program by using the appropriate language and coding approach, testing of the system to ensure every function work properly and debugging the code, which will identify and correct bug within program.

6.1 Development Environment

The initial stage of system implementation involves setting up the development environment. Development environment is very important to the development of a system as suitable hardware and software will determine the success of the project.

6.1.1 Hardware Configuration

The following hardware specifications have been used to develop the system:

- Intel Pentium IV 1.8A GHz
- 512MB DDR RAM
- 40 GB Hard Disk
- 17" color monitor capable of 1024 x 768 resolution
- Standard Input and Output
- Others standard computer peripherals

6.1.2 Software Configuration

There are a lot of software tools, which are used in designing and writing report. Below is a listing of software used throughout the development process as pertaining to the specific usage:

Table 6-1: Software Used

Software	Usage	Description
Microsoft Windows XP	System Development	Operating System
Microsoft SQL Server 2000	Database	Database Management System
Internet Explorer 6.0	System Development	Web Browser
Microsoft Visual Studio .NET	System Development	Authoring Tools
Microsoft Word	System Development	Documentation

6.2 *Platform Development*

Services and tools installations may be the very first step in order to start the development. Platform development includes setting up the operating system and web server.

6.2.1 Setting Up Operating System

Microsoft Windows XP is used as the operating system for this project. Before the installation begins, the hard disk need to be formatted. This is to ensure a more stable and secure environment. Moreover, it can also prevent the environment being affected by previous settings or configurations. Windows XP's installation is very easy as it provides user friendly and descriptive interface guide. User just need to follow the step by step instruction appear on the installation's menu interface.

6.2.2 Setting Up Web Server

Microsoft Internet Information Server (IIS) is chosen as the web server for this project. IIS provides a feature that allows web content to be organized by using virtual servers. It enables user to map local directory to virtual directory and create local web site. Virtual directory is created for InspectionManager.

6.3 Database Implementation

Microsoft SQL Server 2000 is used as DBMS to manage and control database access in InspectionManager. Data retrieving, storing, deleting and other information manipulation activities can be done.

6.3.1 Setting Up Database

After the SQL Server has been installed successfully, a database named 'softspect' is created. The creation of database is done using the SQL Server 2000 Enterprise Manager. After create the database, create the table according to database design. Field types and size of length are specified according to functional requirement and logic.

6.3.2 Database Connection

When building ASP.Net application, connection to database is needed to extract or manipulate data. With ADO.Net, connection to 'softspect' is being done using SQLConnection object. In order to use this object, the first step is to import the System.Data.SqlClient namespace into the application using the "using" keyword.

```
using System.Data.SqlClient;
```

Following is an example of the code for database connection.

```
string connStr = "Data Source=Ahfatt;Initial Catalog=softspect;Integrated Security=SSPI" ;
SqlConnection conn = new SqlConnection(connStr) ;
conn.Open() ;
```

6.4 Program Development

During program development, program is written, user interface is being developed and database is initialized with data.

6.4.1 Program Development Process

Basically, the InspectionManager System is following a program development process that consists of 5 steps:

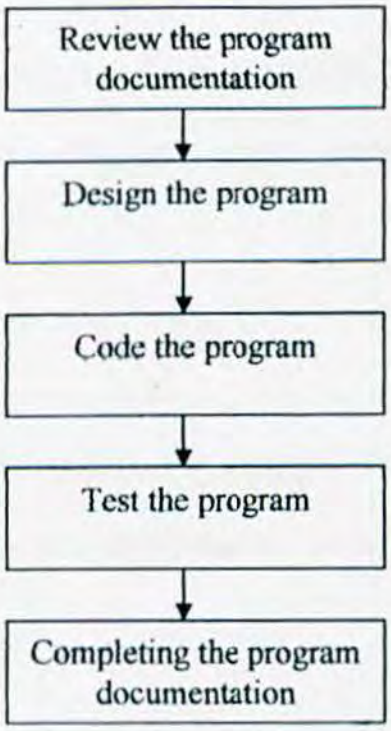


Figure 6-1: Program Development Process

i. Review the Program Documentation

The program documentation that was prepared during the early phases needs to be reviewed. This documentation has the author to understand better of the work that need to be covered during the coding phase.

ii. Design the Program

After review the program documentation, the second level of program design needs to be completed during the system development where the author decides exactly what the program can accomplish. This is the process of what it must do by developing a logical solution to the programming problem.

iii. Code the Program

Coding is a process of writing the program instruction where this instruction implements the program design. The coding step actually translates the design specification to machine-readable format.

iv. Test the Program

During the level program testing, the program processes actual data and produces information on which user will be relying on. The testing involved most are unit testing and integration testing.

v. Completing the Program Documentation

Completing the program is essential for the successful operation and maintenance of the system. This documentation includes the system's user manual that may be needed by most of the customer as well as the system administrator.

6.4.2 Coding Approach

This system is developed modularly using top-down approach. This top-down approach allows the higher-level modules to be coded first before the lower-level modules. The codes in the lower modules contain only an entry and an exit. In shorts, this approach look at the large picture of the system first, and then exploding into smaller part.

6.4.3 Coding Principle Applied

There are a few principles need to apply when coding the program.

Readability

Readability is essential for future enhancement. Coding style and convention applied may strongly affect the readability. Codes need to be formatted to enhance understanding.

Reusability

Reusability is an important principle. It can be considered as a method for improving product quality throughout the system development process. In addition, it also reduces the coding time as well as the testing and documentation time.

Modularity

Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Independent modules are easier to maintain because secondary effects caused by design or code modification are limited, error propagation is reduced, and reusable module are possible.

6.4.4 Style Adopted

The coding paradigm adopted by the system is oriented at giving reliability and performance a balance.

Naming Convention

Naming convention provides easy identification for the programmer. The naming convention is created with coding consistency and standardization in mind. For example:

Textbox	txt
Label	lbl
Button	btn

Indentation and Spacing

The main purpose of indentation and spacing is to ease reading and tracing of code. They make the coding looks neat and tidy.

Program Comments

The comments explain the logic of the certain code, the purpose of a particular program block or other descriptive label. For Example:

```
' Create Instance of Connection and Command Object
string connStr = "Data Source=Ahfatt;Initial Catalog=softspect;Integrated Security=SSPI" ;
SqlConnection conn = new SqlConnection(connStr) ;
conn.Open() ;
string SelectCmd = "select * from login Where user_id = '" + txtID.Text + "' And password = '"
+ txtPass.Text + "'" ;
SqlCommand cmd = new SqlCommand (SelectCmd, conn) ;
SqlDataReader reader ;
reader = cmd.ExecuteReader() ;
```

Sample Code

a. ASP.Net Coding Example

```
<%@ Page Language="C#" Codebehind="default.aspx.cs" AutoEventWireUp="false"%>
```

b. C#.Net Coding Example

i. Insert records into database

```
string user_id, fullname,password,retypepassword,organization,email,projectname, SelectCmd,
InsertCmd ;
user_id = Check_Str(txtUserName.Text) ;
fullname = Check_Str(txtFullname.Text) ;
password = Check_Str(txtPassword.Text);
retypepassword = Check_Str(txtRetypePassword.Text);
organization = Check_Str(txtOrganization.Text);
email = Check_Str(txtEmail.Text);
projectname = Check_Str(txtProjectName.Text); SelectCmd = "Select * From moderator
Where username = '" + user_id + "'" ;
SqlDataReader reader = Create_Rd("moderator", SelectCmd) ;
InsertCmd = "Insert Into
moderator(username,password,fullname,organization,email,projectname) Values('" + user_id +
"','" + password + "','" + fullname + "','" + organization + "','" + email + "','" + projectname + "')"
;
Edit_Data("moderator", InsertCmd) ;
```

ii. Select records from database

```
string ConnectionString = "server=(local); trusted_connection=true; Database=softspect"


SqlConnection MyConnection =New SqlConnection(ConnectionString);
SqlCommand MyCommand  = new SqlCommand ("Select username FROM inspector Where
projectname='"+strProjectname+"'", MyConnection)
MyCommand.Connection.Open()
lblUserName.Text=MyCommand.ExecuteScalar
MyCommand.Connection.Close()
```

iii. Update database record

```
String command="UpDate DEFECTCONFIG Set status = '" + defect1[h] +"'"+
                                    " Where projectname = '" + strProjectname + "'" + "
And defect_type='"+defectType[h]+"'";

Edit_Data("DEFECTCONFIG", command) ;

public void Edit_Data(string DBName, string SQLcmd)
{
string conn_str = "Data Source=Ahfatt;Initial Catalog=softspect;Integrated Security=SSPI" ;
SqlConnection conn = new SqlConnection(conn_str) ;
conn.Open() ;
SqlCommand cmd = new SqlCommand(SQLcmd, conn) ;
cmd.ExecuteNonQuery() ;
conn.Close() ;
}
```

iv. Send mail

```
using System.Web.Mail
:
:
MailMessage Mail = new MailMessage();
```

```
Mail.From = txtFrom.Text;

Mail.To = txtTo.Text;

Mail.Subject = txtSubject.text;

Mail.BodyFormat = MailFormat.Text;

Mail.Body = txtMessage.text;

Mail.Priority = MailPriority.High;

SmtpMail.SmtpServer = "localhost";

SmtpMail.Send (Mail);
```

ASP.NET Validation Control

```
<asp:TextBox id="txtUserID" runat="server" Width="144px" Font-Names="Verdana" />

<asp:RequiredFieldValidator id="validId" runat="server" ControlToValidate="txtId"
ErrorMessage="must do" Font-Names="Verdana" />

<asp:TextBox id="txtPass" runat="server" TextMode="Password" Width="144px" Font-
Names="Verdana" />

<asp:RequiredFieldValidator id="validPass" runat="server" ControlToValidate="txtPass"
ErrorMessage="must do" Font-Names="Verdana" /></P>
```

Chapter 7 System Testing

Testing is critical in uncovering logical error and to test the system reliability. The main objective of testing is to uncover different types of errors that exist while executing the system. System testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. However, testing can only show that software defects are present.

In developing a system, testing usually involves several stages. An example of testing process is shown as below:

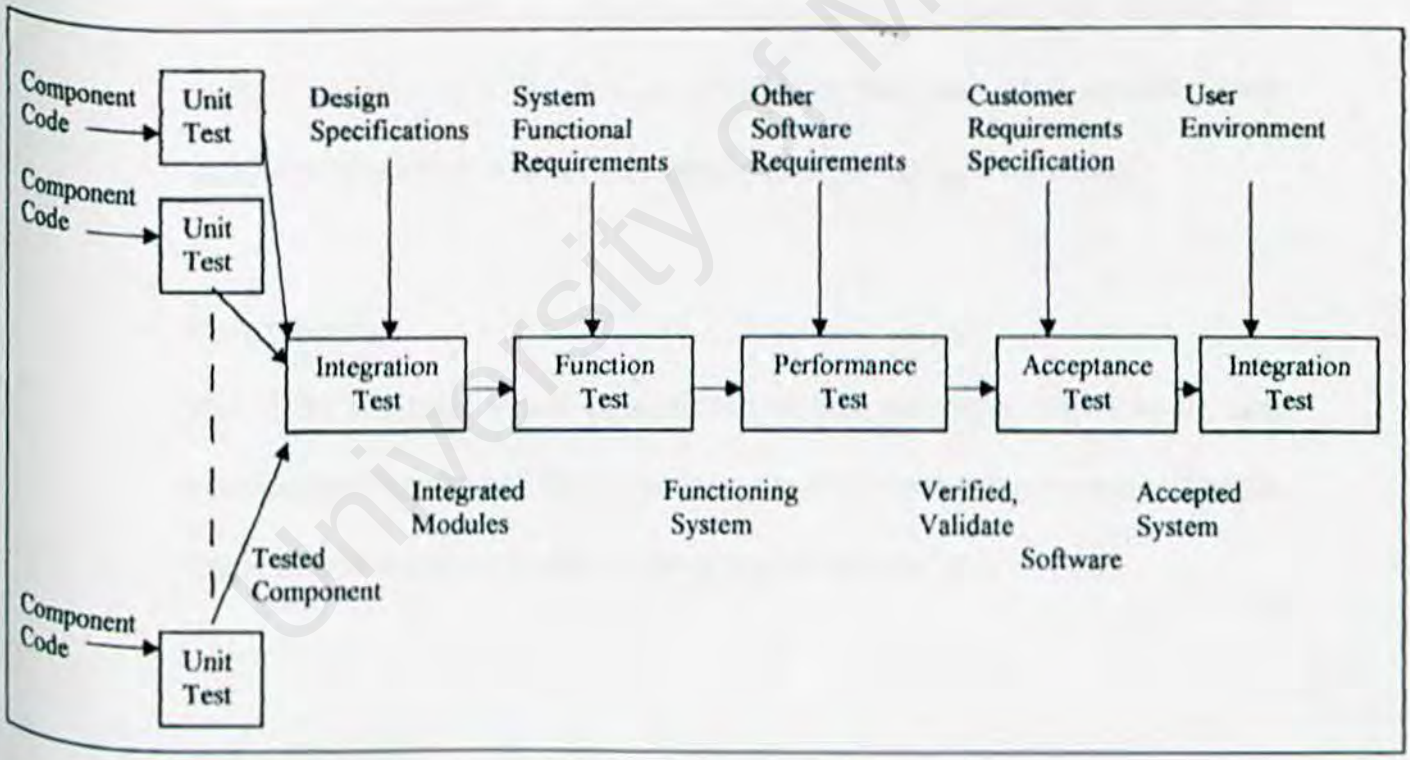


Figure 7-1: Testing process

Generally, there were 3 stages involve altogether and were listed down as below:

a. Unit Testing

This is the first stage of testing where each program component is tested on its own and is isolated from the other components in the system. It verifies that the component functions work properly with the types of input and output expected from studying the component's design. After each component has been tested, the interaction between these components must be tested again to ensure that the components can be integrated.

b. Integration Testing

This stage ensures that the interfaces among the components are defined and handled properly. It is the process of verifying that the system modules work together as described in the system and program design specifications

c. System Testing

This is the last stage which is performed to find out errors, which result from unanticipated interactions of system components or units. It is to ensure that the whole system works according to users' specifications.

7.1 Type of Faults

The types of faults that can be found in InspectionManager are algorithmic faults, throughput faults and computation faults.

7.1.1 Algorithmic faults

Algorithmic faults use to happen when a component's algorithm or logic fails to produce the expected output for a given input. These kinds of faults always are due to wrong proceeding steps. Examples of algorithmic faults in InspectionManager system are:

- Comparison of wrong data type variable
- Forgetting to test for a particular condition.

7.1.2 Throughput/Performance faults

Throughput faults or performance faults use to happen when the component does not produce an expected speed. When discover this fault in InspectionManager system, the fault is being carefully observed and monitor continuously to ensure performance meet requirement. In most cases, the fault occurs during massive usage of graphic on a particular application.

7.1.3 Computation/Precision faults

Computation or precision faults use to occur when a particular formula is being implemented wrongly or the component does not complete a result to a required accuracy. These faults are being tested in component such as appointment booking. In InspectionManager system, all components are being checked and tested with dummy data to ensure the computation works accordingly with its situation.

7.2 Testing Techniques Used

7.2.1 Ad Hoc Testing

Ad hoc testing is an attempt to break the program or make it fail with trying whatever comes to mind. Normally, many errors will be found during the testing.

7.2.2 White Box Testing

White Box testing is the type of testing that deals directly with the stricture of the code within a module or a code segment. There are basically six types of code coverage in white box testing. Most of the testing is discussed in the unit testing (refer section 7.3.1)

Segment Coverage

Each and every segment of the node between control structures is supposed to execute at least once.

Branch Node Coverage

Each and every branch of every possible direction is taken at least once.

Compound Condition Coverage

When multiple conditional appear in the code, every possible combination is tested based on a truth table.

Basis Path Testing

Each independent path through the code is usually taken as predetermined order. When dependencies appears in the code, each path where dependency appears exists must be tested.

Data Flow Testing

In InspectionManager testing, this approach is to uncover anomalies such as variables, which are used but not initialized and declared.

Loop Testing

This type of testing is related to testing single loop (WHILE, FOR LOOP), concatenated loops (sequence of loop) and nested loops (one or more loops within loops).

7.2.3 Black Box Testing

This type of testing involves testing functions of a module without knowing the logic structure of the code. It focuses on the most important aspects of a module in the term of how well the module meets its specification.

Error Guessing

This approach is similar to 'ad hoc testing' where tester will try any type of test cases which come across his/her mind or pre-planned test cases.

Boundary Testing

This type of testing involves the boundaries of equivalent classes where the coverage of test cases will involve inside the boundary, on the boundary and outside the boundary.

Module Interface Testing

In this type of testing, each value within the interface is assured as correct as they related to the modules that call them. This means that specific calls in the calling module are tested to see whether they are in the right sequence and at the right type.

7.3 Testing Strategies

There are a few testing strategies such as unit, integration and system testing are done in order to test the reliability of InspectionManager.

7.3.1 Unit Testing

Unit testing is done to uncover errors in each module. The primary goal of unit testing is to confirm that the unit is correctly coded and that is carries out the function as it is supposing to perform. Each unit is tested independently in order to assure their accuracy. For this system – InspectionManager, each module may contain sub modules and the sub modules may consist of functions. The functions are individually tested before the entire module is tested. In the development of InspectionManager, unit

testing was conducted after development of each of the component and it is a continuous process throughout the coding phase.

InspectionManager Unit Testing

Below are some of the units testing being done on InspectionManager:

- Test whether the user can successfully logged into the Domain Web Server.
- Test whether the data being passed to next program for processing contain the right value.
- Test whether the records being displayed is correct and matches the search criteria.
- After the transaction of data occurs, the related database table are checked to determine whether the insert, modify or delete action has been perform correctly.
- The display is tested after a single program file has been developed to ensure that all the display is correct and expected. These interfaces contain lots of buttons and hyperlinks. Testing for those buttons and links is needed so that the program performs correct action or link to the correct location.
- The program file includes various type of ASP.Net Validation Controls for example RequiredFieldValidator.
- Test whether the send e-mail function can send e-mail to the appropriate receiver stating the correct information.

InspectionManager Debugging Strategies

Debugging is actually of finding and fixing the errors. There are several debugging strategies that applied in InspectionManager such as:

- Built-in Error Detection

  Error will be discovered if a program is not performing well. ASP.NET has built-in error detection where an error message together with the lines number where the error occurred will be debugged. With this features, the debugging work becomes much easier and faster.

- Reviewing the Algorithm Used

  Reviewing algorithm and computations for the correctness and efficiency will help to discover logic error or database error. Usage of different algorithms will sometime increase the efficiency of the program.

- Display the Passing Value On Screen

  By displaying the passing value on screen, it helps to ensure that the correct value has been passed to the next program for processing.

- Check Success Status

  The success status is checked to determine whether to continue the process or exit from the program and display error message whenever there is failure in the previous process.

- Using Query Analyzer Provided by SQL Server 2000

  Query analyzer will helps to test the SQL statement and information about the error will be provided. Query Analyzer is also used to correct the SQL statement when wrong information is being retrieved.

7.3.2 Integration Testing

The purpose of the integration testing is to know whether the entire software is able to work as one program. It will also verify that each module will be able to function together. Integration testing concentrates on module interaction and the detection of interface errors. The design specification is referred for the purpose of verification and helps to test the software according to the dependencies present in particular module that being tested. For InspectionManager integration testing, the system is viewed as a hierarchy of components, where each component belongs to a layer of design. The approach applied in testing the InspectionManager system is referred as Top-Down Integration where integration will start at the highest level of main program or module or sub modules are gradually added until the bottom is reached.

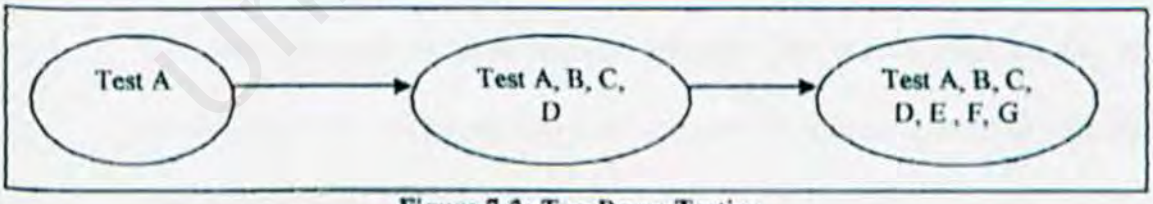


Figure 7-2: Top-Down Testing

7.3.3 System Testing

A system testing is a series of different test designed to fully exercise the system to uncover its limitation and to measure its capabilities. The objective is to test an integrated system and verify that it meets the specified requirements. Several steps were taken in testing InspectionManager system such as function testing and performance testing.

Function Testing

System testing begins with function testing that focus completely on functionality. The system structure is being ignored. The testing is based on the system's functional requirements which are stated in the early chapter.

Performance Testing

Performance testing aims at testing the run-time performance. Response time of the event triggered was checked to verify the performance of the system.

a. Stress Testing

   The main purpose is to determine whether the system can handle, as it should, large and varies workload at one time. It subject system to high loads over a short period of time.

b. Security Testing

The objective is to verify the protection mechanism such as dealing with improper penetration.

c. Human Factor Testing

In this testing, interface and message are being evaluated by many user to get the best interaction effects. It concentrates at the appearance and the interaction of the system. All aspect that may be related to ease of use, such as display screen, will be examined.

7.4 Test Cases

Test cases are developed to show that the input is properly converted to the desired output. They are used as some set of structural input is given and the output is observed. The test cases are design to perform unit testing till integration testing with the specific results. Repetitive testing is done on a single test case to prove the consistency of the results.

7.4.1 Unit Test Cases

Unit test cases aim to test an individual independent component. Below is an example of unit test case that being done for InspectionManager.

Test Case	: 4
Module	: Inspector
Sub-module	: Log Defect

Unit : Defect info – Delete, Edit Scenario : To view and delete a defect's information				
No	**Steps/Fields**	**Test Data**	**Expected Result**	**Test Result**
1	Click "Log Defect" button.		System displays a datagrid contains defect's information	Defect's information displayed.
2	Click Delete button		System displays message box to confirm deletion	Message box displayed.
3	Click Cancel button on message box		System closes the message box	Message box closed
4	Click Delete button again		System displays message box to confirm deletion	Message box displayed.
5	Click Ok button on the message box		System deletes the specific record and update the record	Record deleted and updated record displayed
Status : Pass Date : 10/12/2004				

Figure 7-3 : Unit Test Case

7.4.2 Integration Test Cases

Integration test cases involve more than one component. It usually tests for a sequence of flows and aims at verifying the interface between different components. Below is an example of integration test case for InspectionManager system:

Test Case	: 2
Module	: Moderator
Sub Module	: Authentication, Defects Merging
Unit	: Authentication, Merge Discovery Logs, Select Meeting Defects
Scenario	: Moderator will log in, merge the discovery logs from all the inspector and select the most possible defects to be discuss in the meeting.

No	Steps/Fields	Test Data	Expected Result	Test Result
1.	Enter the URL		System displays main page	Main page displayed
2	Click Login button on main page navigation bar		System displays Login page	Login page displayed
3	Enter invalid username & password.	Test, 123	System denies the access and prompt error message	Access denied and error message prompted
4	Enter valid username & password	viva, viva	System allows the access and redirect to moderator page	Access granted and member page displayed
5	Click "Merge Discovery		System displays	Page display

	logs" button		instruction and button to merge now.	succesfully
6	Click "Merge now" button		System display all the high potential defects in datagrid	Defects are display properly
7	Click "continue"		System direct moderator to "select meeting defects" page	select meeting defects page displayed
8	Moderator check the defects desired.	check data number 1,2,3	Record number 1,2,3 send to log formal defect page	Three records display correctly.
9	Click "Back to menu" button		System display moderator's menu page	Moderator menu page displayed
Status : Pass Date : 22/12/2004				

Figure 7-4: Integration Test Case

Chapter 8 System Evaluation

Evaluation is the ultimate phase of developing a system and an important phase before delivery the system to the end user. System evaluation is implemented by more than simply comparing the information obtained with the information which is expected. It was related to user environment, attitudes, information priorities and several other concerns that are to be considered carefully before effectiveness can be concluded. At all phases of the system approaches, evaluation is a process that occurs continuously, drawing on a variety of sources and information.

8.1 Problem Encountered and Solution

The following are the major problems encountered from the beginning of the project through the end of the system development process.

8.1.1 Scope Is Not Properly Defined

In the initial stage of development, several problems were encountered in specifying the scope of the system. The scope must be clearly defined before the coding can start. The solutions to solve the problems are through interviewing the potential users, checking the current available similar site and also having discussion with supervisor. Before the viva, the scope for this project was too wide and the project moderator, Dr. Lee advises to narrow down the scope to make it concentrate at requirement document.

8.1.2 Problems in Tools and Language Selection

There are many types of Web based system development tools available nowadays. Choosing the right tools and language are important because the appropriate tools and language would help to develop the system in a more efficient way. The solution taken including seeking advice from supervisor, discussion with course mates who develop similar system and also having some research and review on various tools and language. The Internet BBS group also provides a lot of suggestions.

8.1.3 Inexperienced In the Chosen Programming Language

Due to time constraint, the learning and developing process was done in parallel. Since there was no prior knowledge of programming in ASP.Net, there was an uncertainty on how to develop a web based system using ASP.Net. Although having difficulty in the early stages, however choosing to program in ASP.Net proves to be a wise move as it is a very powerful technology to build a web based application. Problems were solved through research on related material online and referring to some reference books. Discussion with friends using similar technology also helps to solve some difficulties.

8.1.4 Difficulties in Designing User Interface

web based system interface, it helps to design the user interface in a more presentable and attractive style.

8.1.5 Limited Knowledge about Software Inspection

Limited knowledge about Software Inspection causes difficulties in developing the system. Problem solved by research in library and also searching information via internet.

8.2 System Strengths

The system strengths are described as follow:

8.2.1 Proper Identification And Authentication

The security procedures in the login code make sure that only authorized person are allowed to enter the specific page.

8.2.2 User Friendliness

As a web based system, InspectionManager shows some of the advantages in term of its usability. Consistent user interface are available in this system. InspectionManager provides a standard interface appearance through the whole system. Besides, it has friendly GUI where all type of button is well defined. This is to ensure that the user can easily use the system without any briefing or with minimal training.

8.2.3 Ease Of Getting Information

The system provides various of information such as checklists and also guidelines about the inspection process. User can get all these information by just a few mouse click.

8.2.4 System Transparency

System transparency refers to the condition where the users do not need to know about the structure, where the database resides, its database management system and anything related to the system implementation. For instance, users do not need to know how to retrieve and insert records into database and how to update their information. All they need to do is to submit data required and then view the results.

8.2.5 Web-site Content Management

Staffs are allowed to perform their task on-line, for example update, insert and delete records.

8.3 System Constraints

The system constraints are described as below:

8.3.1 Limited Functionality

The author correction on the defects found during meeting is absence in the system.

8.3.2 Only Support English As single Communication Language

The system will only use English as single communication language as English is the international language. Other languages are not included in the system.

8.4 Future Enhancement

Due to the limitation of this system, there are a few suggestions that may be useful to future enhancement of the InspectionManager system. The suggestions are as below:

8.4.1 More Functionality Added

Extend the functionality of the system that enables the author to correct the defect found and reply on the defects.

8.4.2 Provide Forum Function

Forum can be integrated into this system so that moderator can leave messages to inspectors and create threads to discuss the defects found. This can save more time and *is possible finish the inspection without have to organizing the inspection meeting and make the inspection capable to support distributed place inspection.*

8.5 Knowledge And Experience Gained

Besides knowledge on technical aspects such as Windows XP Server, ASP.Net, C#.Net and SQL Server, there are also other valuable experiences gained from working on this project such as:

- Being exposed to the real system development environment especially dealing with users
- Learn how to manage a project as in time and resource
- Concept on how to integrate and fully utilize various technologies into developing system
- Experience on how to set up and configure various technologies to be able to serve as a live system.
- Learn to work independently
- Cultivated skills in writing documentations and reports
- Boost self-confidence, self-esteem and good communication skill

8.6 Reviews on Goals

There should be certain expectation and objective achieved at the final stage of the project.

8.6.1 Expectation Achieved

The system had fulfilled the expectation stated at the early stage of the project. All the basic foundation of the system was being designed and implemented. Moreover, the end product met the criteria such as user friendliness, reliability, manageability, expandability and so on.

8.6.2 Objective Achieved

The system created had fulfilled all the requirements stated in the early chapter, therefore, the objectives to establish the application had been achieved.

8.7 Chapter Summary

As a conclusion, this project was succeeded in achieving the objectives of developing a web based software Inspection system. It also projected the main idea of general office environment as to promote a paperless environment with the routing of information through the workflow application.

Throughout the development of this project, a lot of precious knowledge on web based programming was gained. This included the configuration and management of Windows XP Server and IIS, programming knowledge in ASP.Net and C#.Net as well as the techniques and concepts in implementing database (SQL Server). This project has been a very useful experience which exposes the idea of research work to the developer.

REFERENCE

[1] M. Fagan. *Design and code inspections to reduce errors in program development.* IBM System Journal, 15(3):182-211, 1976.

[2] J. M. Perpich, D.E.Perry, A. A. Porter and L. G. Votta. *Anywhere, anytime code inspections: using the Web to remove inspection bottlenecks in large-scale software development.* International Conference on Software Engineering, Proceedings of the 19th international conference on Software engineering. Pages 14-21, 1997

[3] Schach R,Stephen (2005) Object-Oriented and Classical Software Engineering. New York, McGraw Hill, Inc.

[4] B. Boehm. *Industrial Metrics Top-10 List.* IEEE Software, September 1987, 84-85.

[5] Ebenua, Robert. (1994). *Software Inspection Process.* New York: McGraw-Hill, Inc.

[6] Strauss, S. H. and R. G. Ebenau (1994) *Software Inspection Process,* New York: McGraw-Hill, Inc.

[7] Gilb,T. and D.Graham (1993) *Software Inspection.* New York: Addison-Wesley Pub Co.

[8] National Aeronautics and Space Administration. *Software Formal Inspections Guidebook,* NASA-GB-A302. August, 1993.

[9] National Aeronautics and Space Administration, Software Formal Inspections Standard, NASA-STD-2202-93, April 1993

[10] Macdonald, F. and J. Miller, *A Comparison of Tool-Based and Paper-Based Software Inpection,* URL: http://www2.umassd.edu/SWPI/ISERN/ISERN-98-17.pdf

[11] Whitten.J,Bentley,L,and Dittman,K (2004) *System Analysis and Design Methods* (6th Edition) New York: McGraw Hill, Inc.

[12] http://chinese.cari.com.my/myforum/viewthread.php?tid=237760
Last Visit: 25-1-2005

[13] http://chinese.cari.com.my/myforum/viewthread.php?tid=216585
Last Visit: 14-12-2004

[14] http://chinese.cari.com.my/myforum/viewthread.php?tid=215094
Last Visit: 14-12-2004

InspectionManager v1.0
User Manual

Table of Content

List of Figure

Chapter 1 Introduction

InspectionManager is a web based system target at the user group which has knowledge in software inspection process, and specially designed to help inspection Moderator. The system will allow the Moderator to control on his inspection group, select/invite his inspection team, make analysis on the defects found and log the inspection defects. For the inspector which been invite to join the inspection, they are manage to view/download the requirement document and log defects.

1.1 About This Manual

This user manual will guide you through all the function available in the system. This manual includes the following part such as:

- System Overview
- Hardware and Software Requirement
- Moderator Module
- Inspector Module

1.2 Conventions

To help you to locate and interpret information easily, this user manual uses the consistent typographic. These conventions are explained as follows:

- [Button] : Indicates a button in the system
- **Menu/Option** : Indicates menu bar in the system
- Hyperlink : Indicates a hyperlink in the system

Chapter 2 Hardware & Software Requirements

2.1 Server Side Requirements

2.1.1 Hardware Requirements

Hardware requirements for the server computer are:

- PC with a Pentium (233 MHz or higher)
- At least 128 MB of RAM (256 MB is recommended)
- Hard disk space of 2.5 GB
- VGA or higher-resolution monitor, super VGA recommended
- Others standard computer peripheral

2.1.2 Software Requirements

The software required to be installed into the server computer are:

- Microsoft Windows XP
- Microsoft SQL Server 2000
- Microsoft Internet Information Server 5.0 with .NET framework
- Internet Explorer 6.0

2.2 Client Side Requirements

2.2.1 Hardware Requirements

The hardware requirements to execute the system are listed below:

- PC with a Pentium (133 MHz or higher)
- At least 64MB of RAM (128 MB is recommended)
- Hard disk space of 2.5 GB (recommended)
- A VGA or other compatible monitor display
- Others standard computer peripheral

2.2.2 Software Requirements

The software required to be installed into the client computer are:

- Any platforms and browser that suitable(Internet Explorer 5.5 is recommended for browser)

Chapter 3 Getting Started

3.1 Setting up InspectionManager Virtual Directory

1. Start **Windows Explorer** and create a new physical directory named "softspect", under the \Inetpub\wwwroot directory created by IIS on your hard drive.
2. Copy all the softspect files to this directory
3. Then, click **Start,** point to **Control Panel**, point to **Performance and maintenance**, point to **Administrative Tools**, and then click **Internet Information Services**
4. Expand <**domain name**> by clicking +. Right click on **Default Web Site** and point to **New**, select **Virtual Directory**. You will see the splash screen of the **Virtual Directory Creation Wizards**. Click on [Next]
5. Type softspect in the Alias text box; then click on [Next].
6. On the next screen, click [Browse] and select the directory **Inetpub/wwwroot/softspect** that you have created in step 1. Then, click [Next].
7. Make sure that the **Read** and **Run** scripts checkboxes are checked, and that the **Execute** checkbox is empty. Click on [Next] and the click on [Finish].

The softspect virtual directory will appear on the tree in the IIS Administration Windows as shown in Figure 3-1.

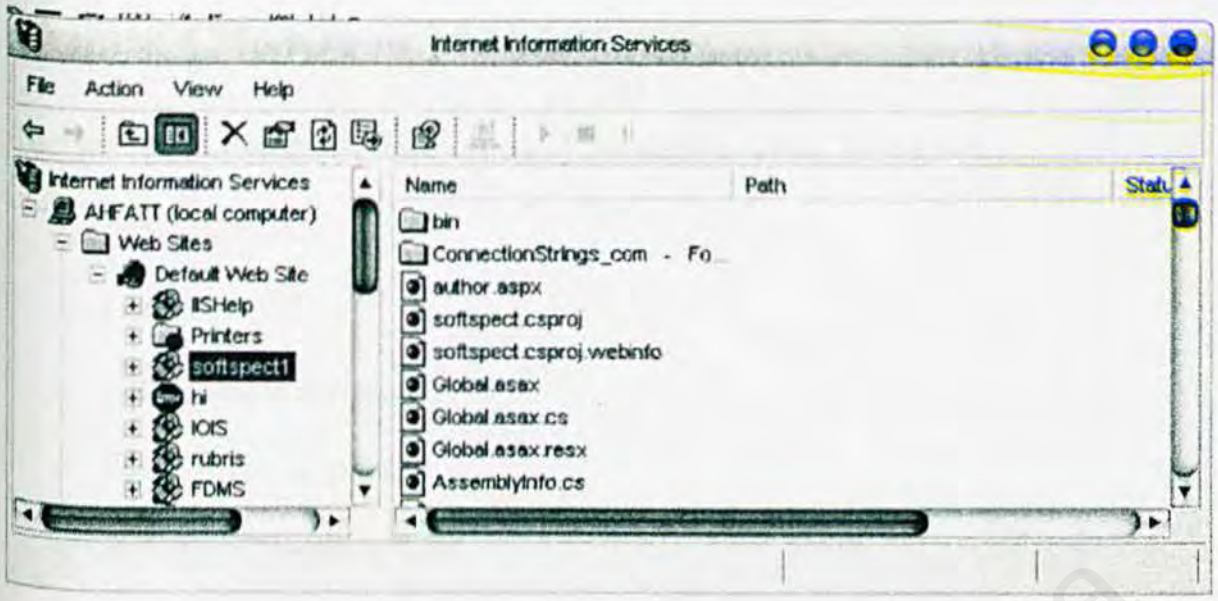


Figure 3-1: Internet Information Services

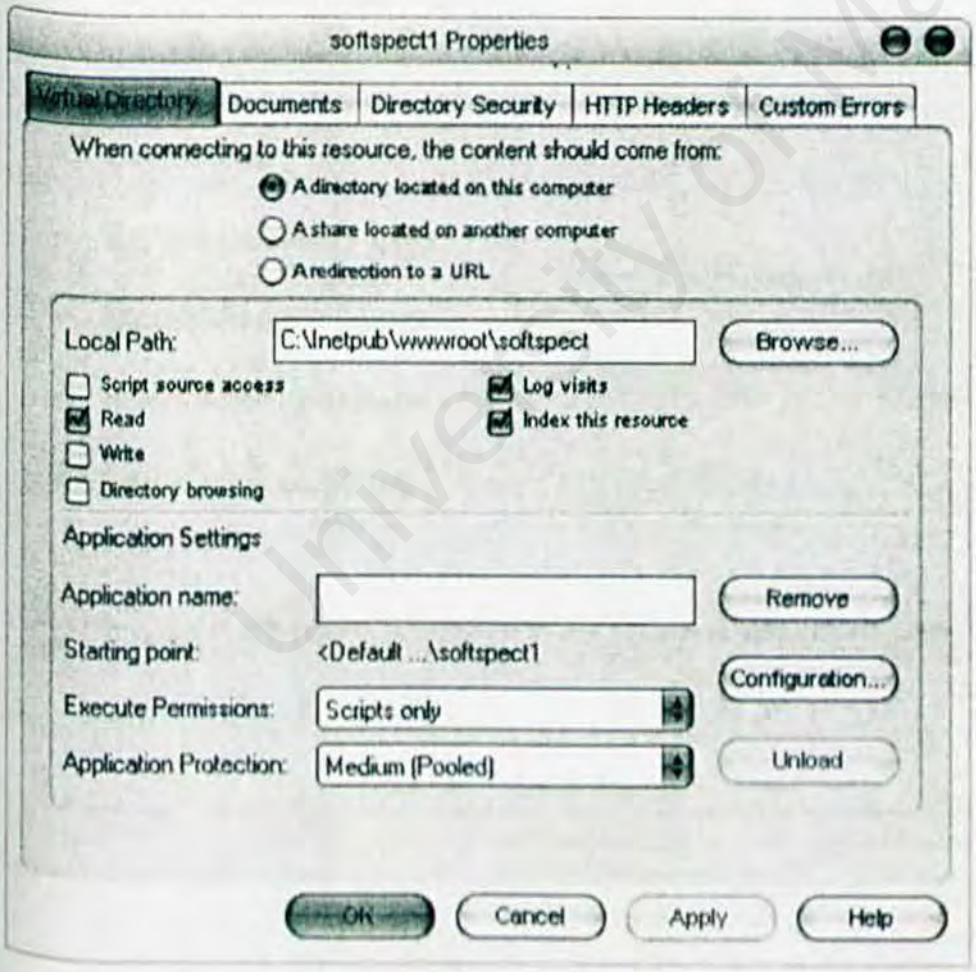


Figure 3-2: Softspect Virtual Directory

Chapter 4 Moderator Module

This module contents the description of the user function, which are:

- New Inspection
- Log in
- Customize Inspection
- Select/invite Inspector
- Schedule Overview Date
- Schedule Meeting Date
- Upload Document
- View Inspection Information
- Review Document
- Log Defects
- Review Discovery Logs
- Merge Discovery Logs
- Select Meeting Defects
- Log Advanced defects
- Print
- Logout

4.1 User Main Page

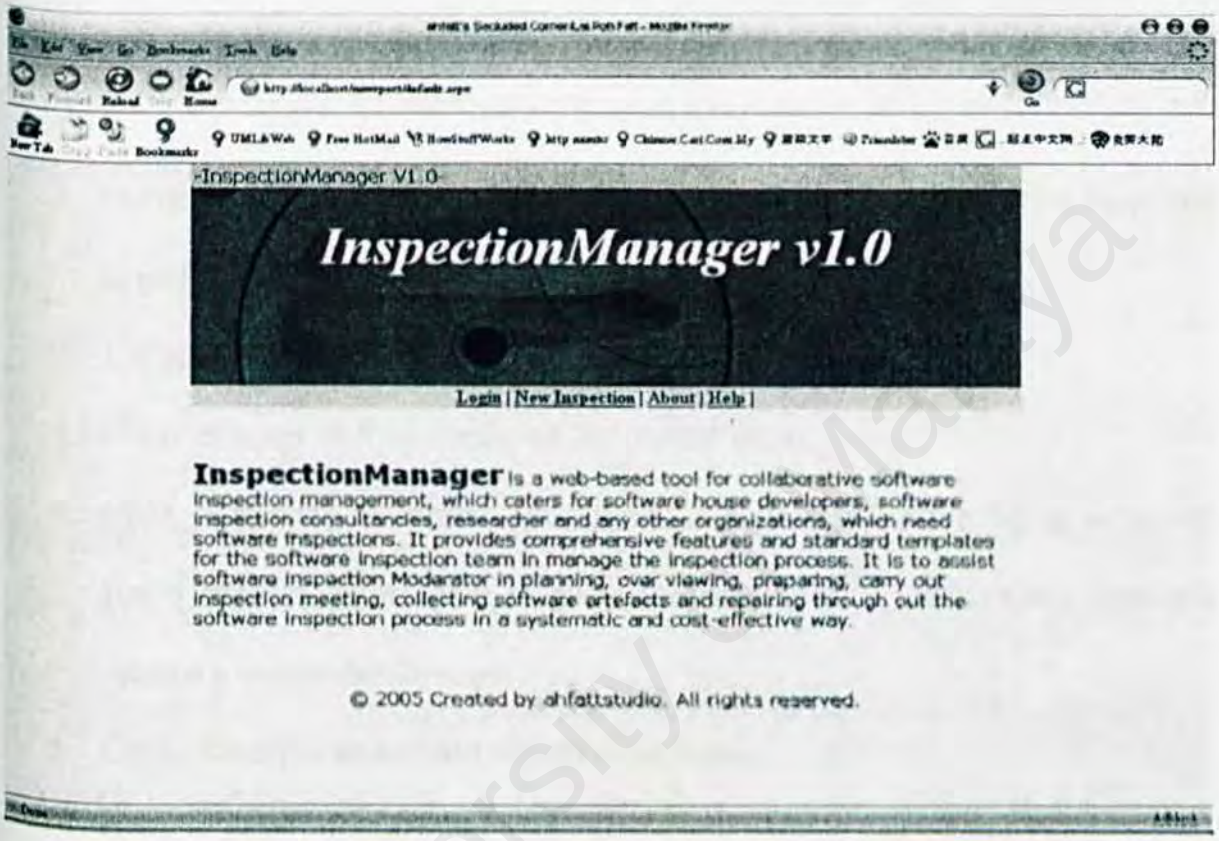


Figure 4-1: User Main Page

Getting started:

1. The system website can be accessed through the URL address: http://YourServerName/softspect/default.aspx
2. The **User Main Page** consists of 4 buttons which will link to some general page or function about the administration of the system.
3. The function of the buttons are as below:
   - [Login] enable member to log in and perform their task

- [New Inspection] enable moderator to create a new inspection project.
- [About] bring you to the system's information.
- [Help] bring you to the system's help file.

4.2 New Inspection

If you wish to create a new inspection, you will need to register with the system

1. Move your mouse to [New Inspection] and click it. The registration form will appear as shown in figure 4-2.
2. Fill in all the required field as mentioned.
3. Error message will be displayed for invalid input.
4. Click [Register] to submit your form. If your registration being accepted, you'll need to login again to go to the moderator's menu. Else, your will receive a failure notification.
5. Click [Reset] to re-do your registration form.
6. Click [Cancel] to cancel the registration and back to **User Main Page**.

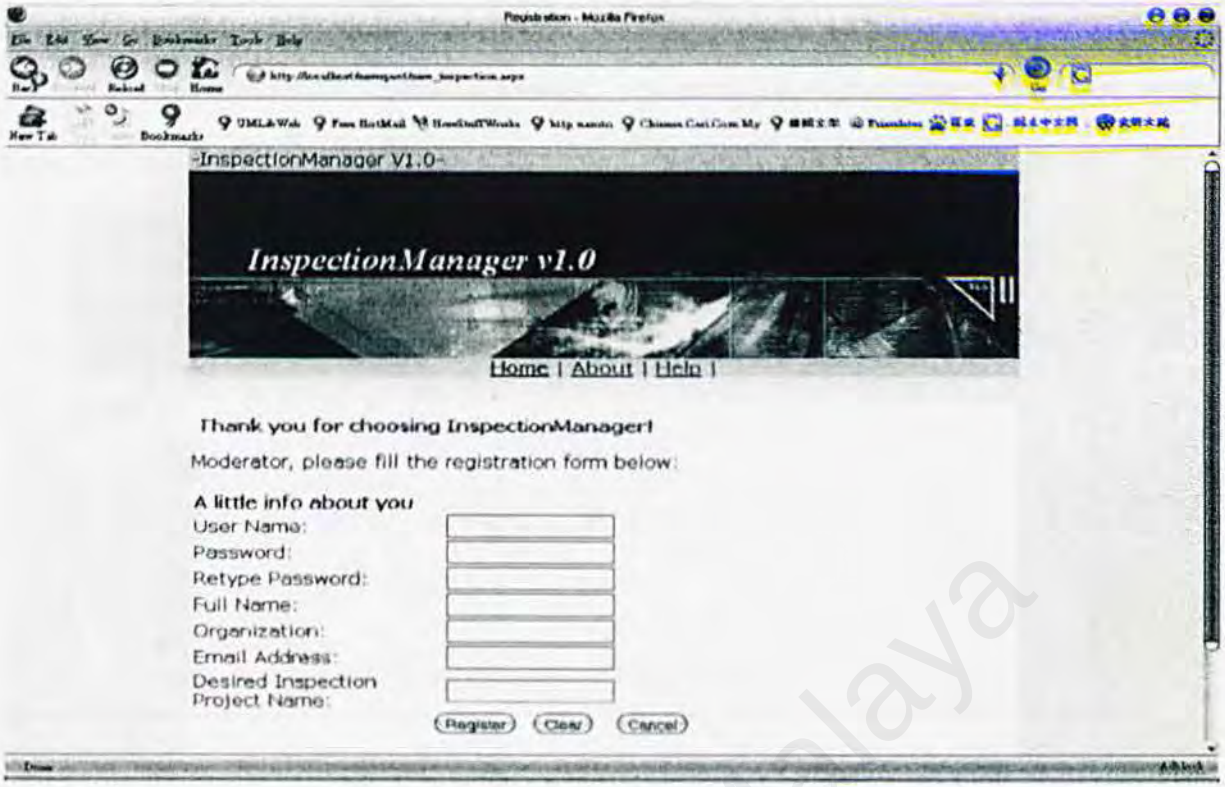


Figure 4-2: New Inspection Registration Form

4.3 Moderator Log In

In order to ensure the security and enjoy all the services provided, member is required to login as an authorized user.

1. Move your mouse to [Login] and click it. The **Login Screen** will appear as shown in figure 4-3.
2. Enter your user name and password and your roles.
3. Click [Login] to log in. Error message will be displayed for invalid user name and password and you will need to re-enter your user name and password. After you have log in successfully, you will be redirect to **Member Page** as shown in figure 4-4

4. Click [Cancel] to go back to **Main Page**.

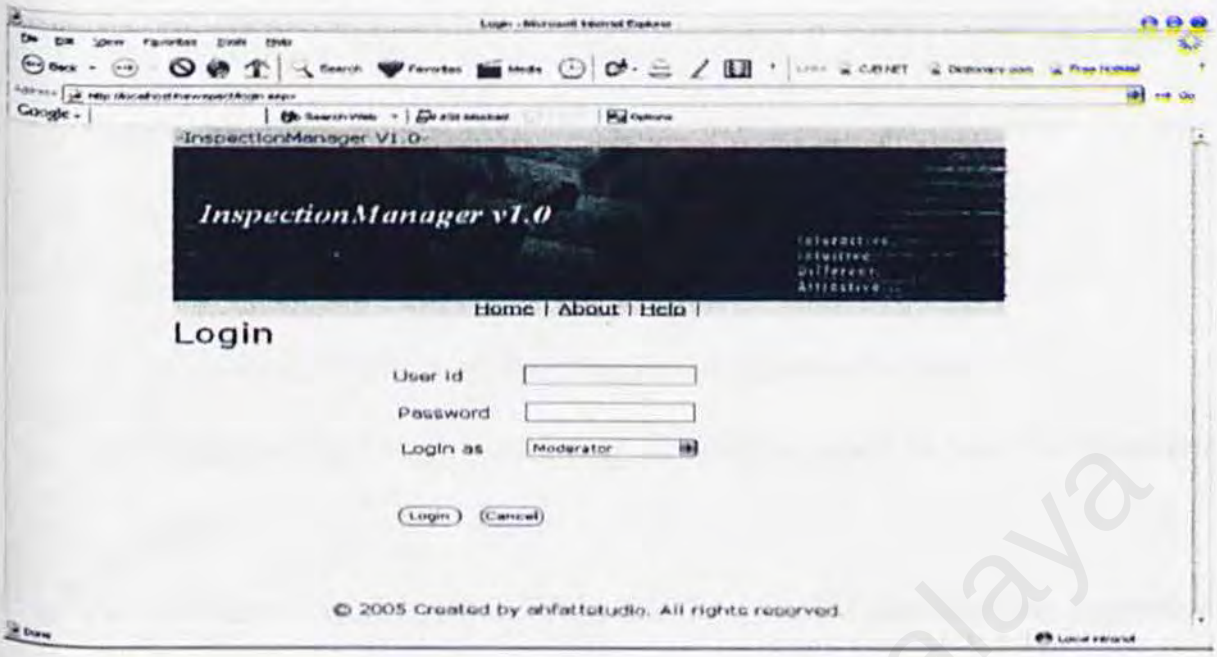


Figure 4-3: Member Log In

Figure 4-4: Moderator Page

4.4 Moderator Page

1. The **Moderator Page** (refer to figure 4-4) is the main menu for all the function provided.
2. The page consists of 14 vertical buttons and 4 horizontal buttons
3. Vertical Buttons:
   - [Customize Inspection] Specified your inspection's detail.
   - [Select/invite Team] Invite your desired inspector to join the inspection team.
   - [Schedule Overview Date] Schedule the overview date for your inspection
   - [Schedule Meeting Date] Schedule the meeting date for your inspection.
   - [Upload Document] Upload the requirement document.
   - [Information] View Inspection's information.
   - [Review Document] View the selected document to be inspected
   - [Log Defects] Log the potential defects discovered to be discuss in inspection meeting.
   - [Review Discovery Logs] Review each inspector's defect log to check their effort
   - [Merge Discovery Logs] Analyzed all the defects and merge in a list which the defects are discovered more than one inspector
   - [Log Prepared Defect] Log the defect according to be well prepared defect before meeting
   - [Log Advanced Defect] Log advance defects discovered during meeting.
   - [Print] Print Necessary documents.

- [Review Document] View the selected document to be inspected

4. Horizontal Buttons:

   - [Home] bring you to the main page.
   - [Logout] enable you log out of the system and back to main page.
   - [About] bring you to the system information
   - [Help] bring you to the system's help Page

4.5 Customize Inspection

This guide will help you to use all the function about **Customize Inspection**

1. Move your mouse to [Customize Inspection] and click it. Your inspection customization page will appear as shown in figure 4-5.
2. Select the type of severity and type of defect that you wish to discover in this inspection. Check on the checkbox "select all" will select all the items listed.
3. Click [OK] to save your setting or a click on [Back to menu] will bring you back to Menu page.

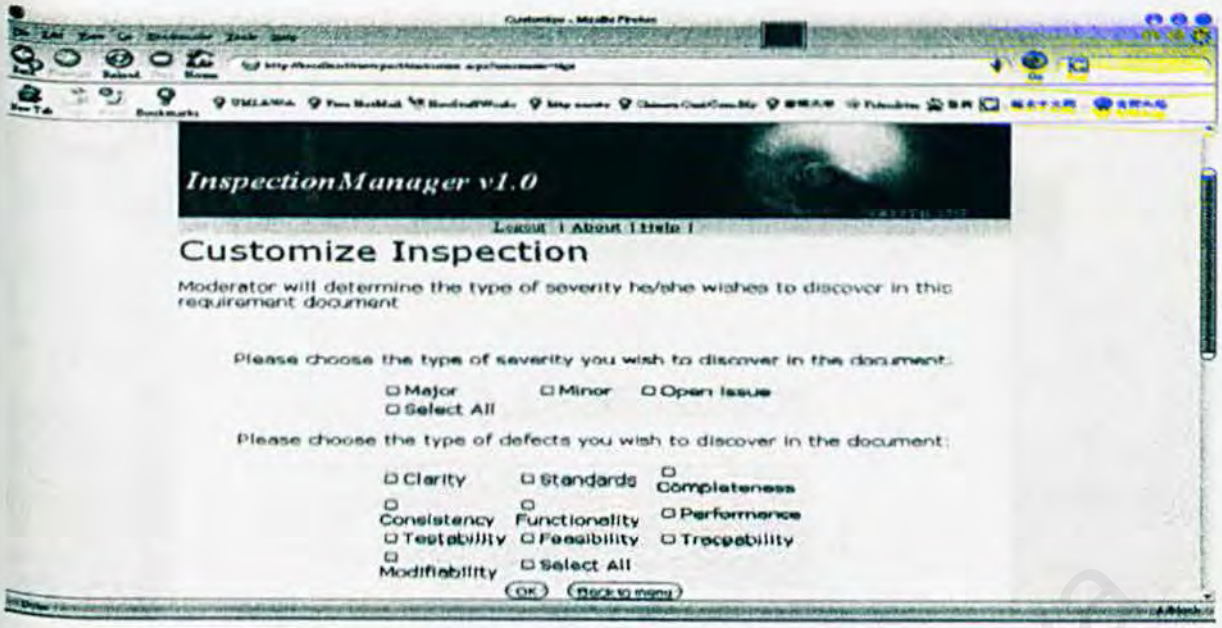


Figure 4-5: To Customize Inspection

4.6 Invite Inspector

This guide will help you to use all the function in **Invite Inspector**.

1. Move your mouse to [Select/Invite Team] and click it. Your information will appear as shown in figure 4-6.
2. Fill in the recipient, your email, subject besides additional information and click on the [Invite Now] button.
3. Your invitation is sent and system will bring you back to the Menu page.

Figure 4-6: Invite Inspector

4.7 Schedule Overview/Meeting Date

This guide will help you to use all the function in **Schedule Overview/ Meeting Date.**

1. Move your mouse to [Schedule Overview Date]/ [Schedule Meeting Date]/ and click it. Your information will appear as shown in figure 4-7.
2. System will show you the previous date you choosed before. If you're first time enter hear, it will show "null".
3. Select the date from the calendar showed.
4. Fill up the venue column, choose the time and fill up the message column.

5. Click on [Save] button to save your information and system will directed you to the Menu page.

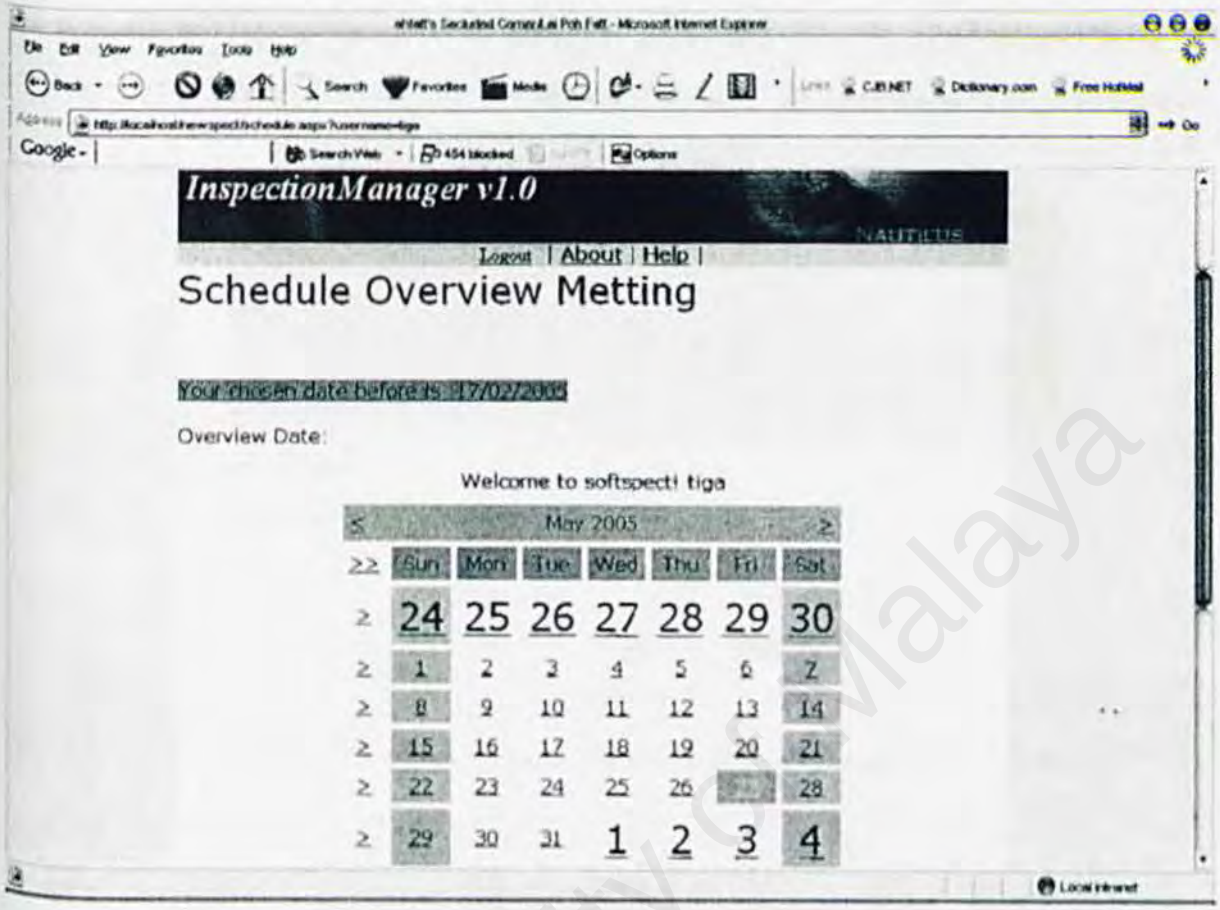


Figure 4-7: Schedule Overview/Meeting Date Page

4.8 Upload Document

This guide will help you to use all the function in **Upload Document.**

1. Move your mouse to [Upload Document] and click it. Your information will appear as shown in figure 4-8.
2. Click [browse] and select your file to upload.
3. Click at [Upload File] to upload the desired requirement document while click at [Cancel] will bring you back to menu.

4. If upload successful, your file name, file type, file size and file location will shown by the system.

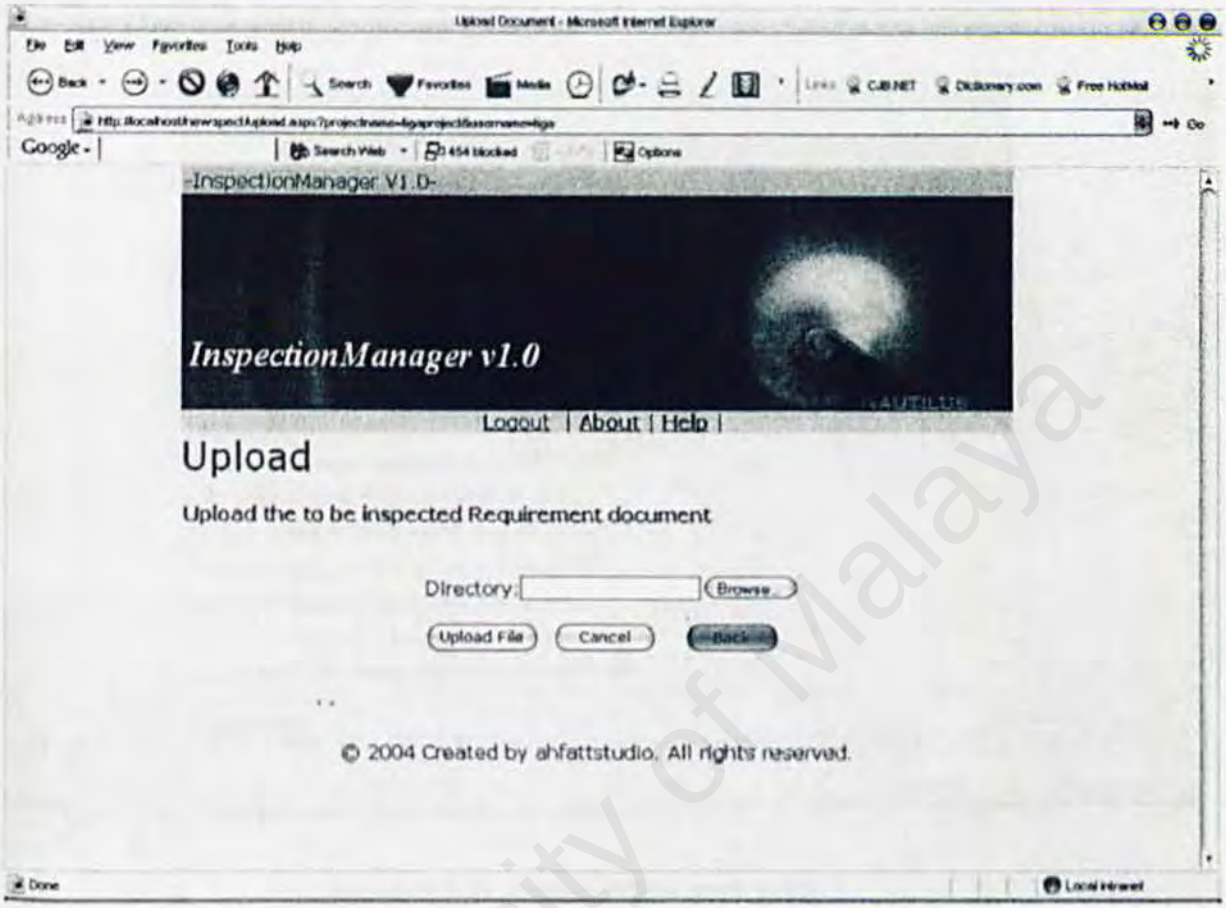


Figure 4-8: Upload Document

4.9 View Information

This guide will help you to use all the function in **Information Page.**

1. Move your mouse to [Information] and click it. Your information will appear as shown in figure 4-9.
2. Information related to the inspection project : Project name, Moderator's Name, Moderator's Email, Overview's time, venue and date, Meeting's time, venue and date, and document to be inspected are shown.

3. Click on [Back to menu] will bring you back to menu page

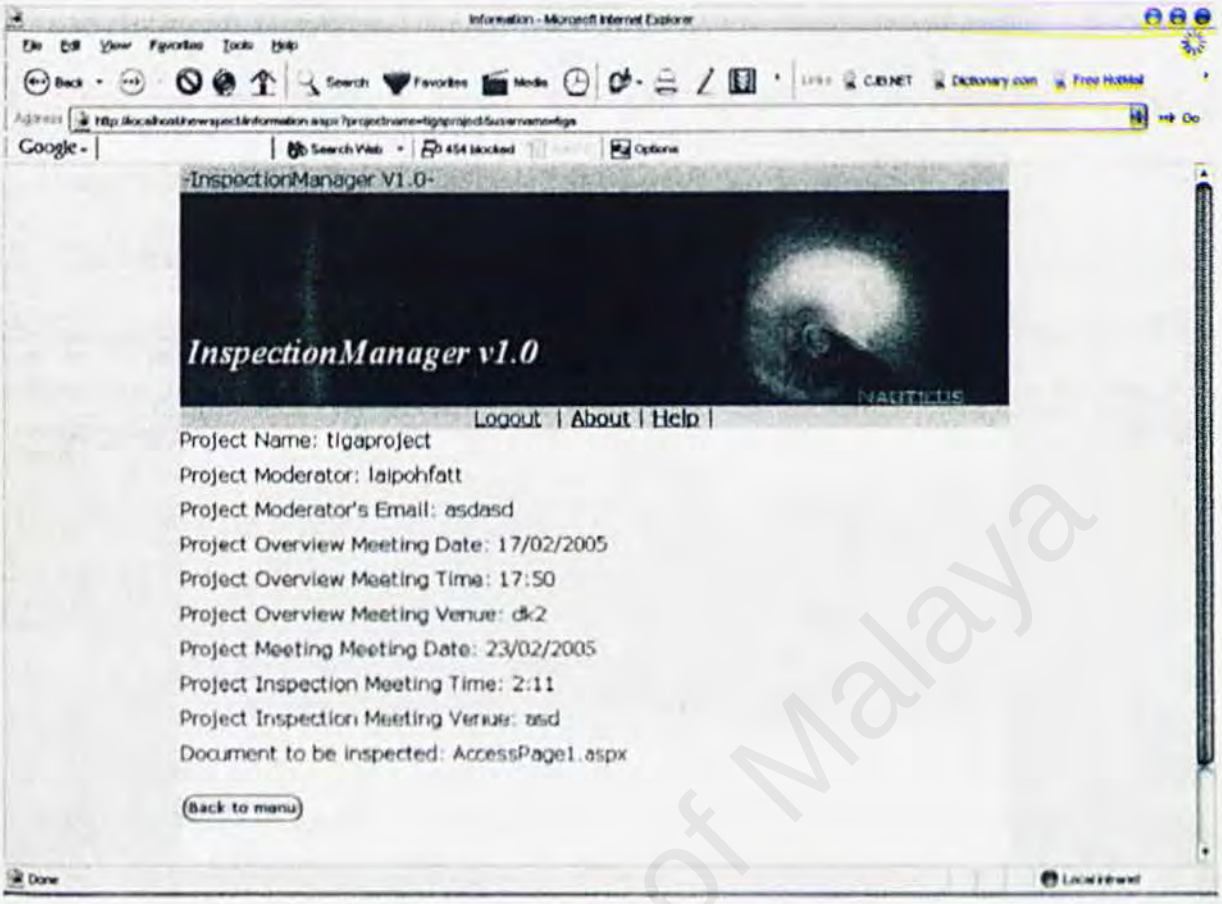


Figure 4-9: Inspection Information page.

4.10 Review Document

This guide will help you to use all the function in **Review Document.**

1. Move your mouse to [Review Document] and click it. Your information will appear as shown in figure 4-10.
2. Your document to be inspected will be shown by the system.

3. Click on the [Requirement Checklist] will pop-up a windows contain the requirements checklist.
4. Click on the [Review Now], you can review or download/save the requirements document.
5. Click on the [Back to Menu] will bring you back to the Menu page.

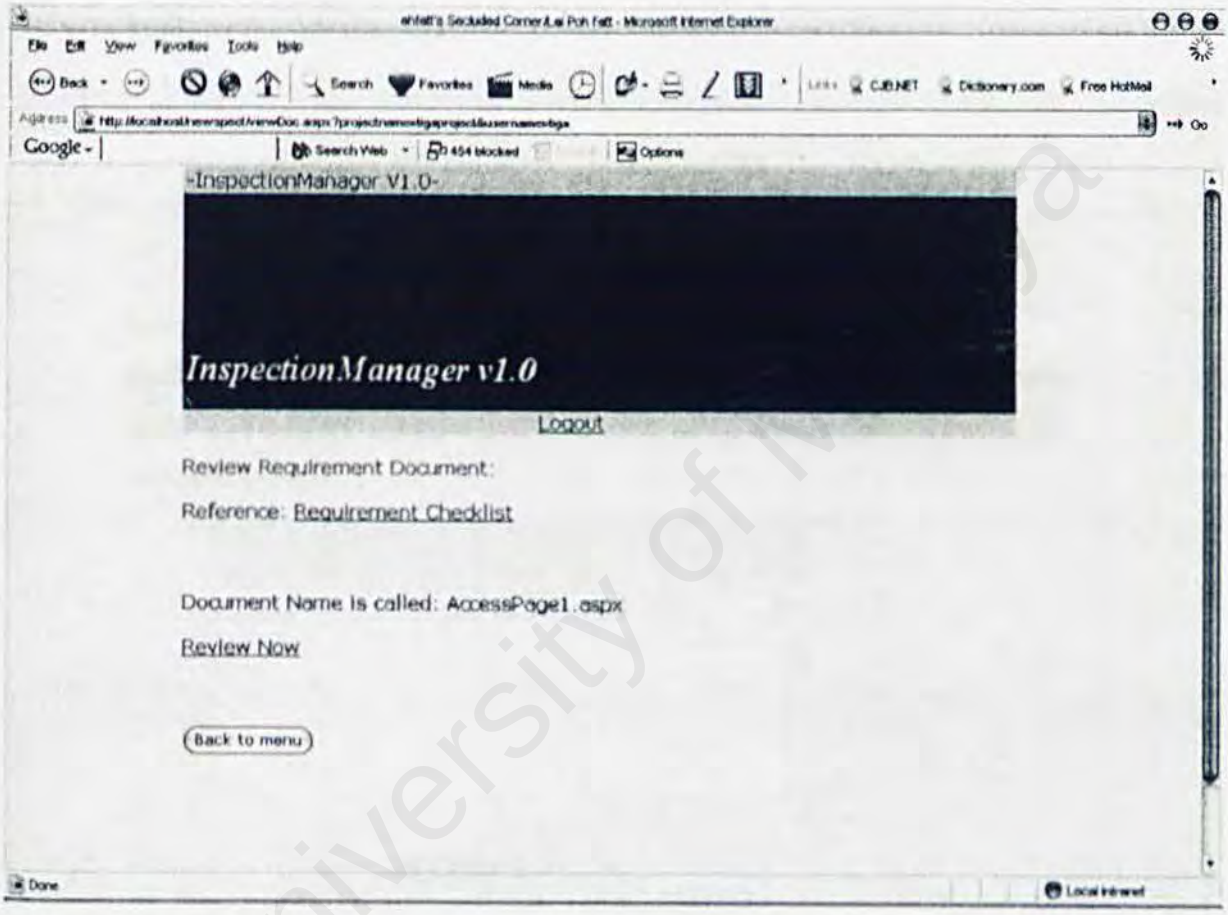


Figure 4-10: Review Document page.

4.11 Log Defects

This guide will help you to use all the function in **Log Defects.**

1. Move your mouse to [Log Defects] and click it. Your information will appear as shown in figure 4-11.

2. System will display a table contain the previous logged defects. (Empty if no record was logged previously)
3. To add a new defect, fill in the location, page, select the type of defect, and select the type of severity and fill the comment column.
4. Click [Log defect] to save your entry while click [Back to menu] will bring you back to the Menu page.

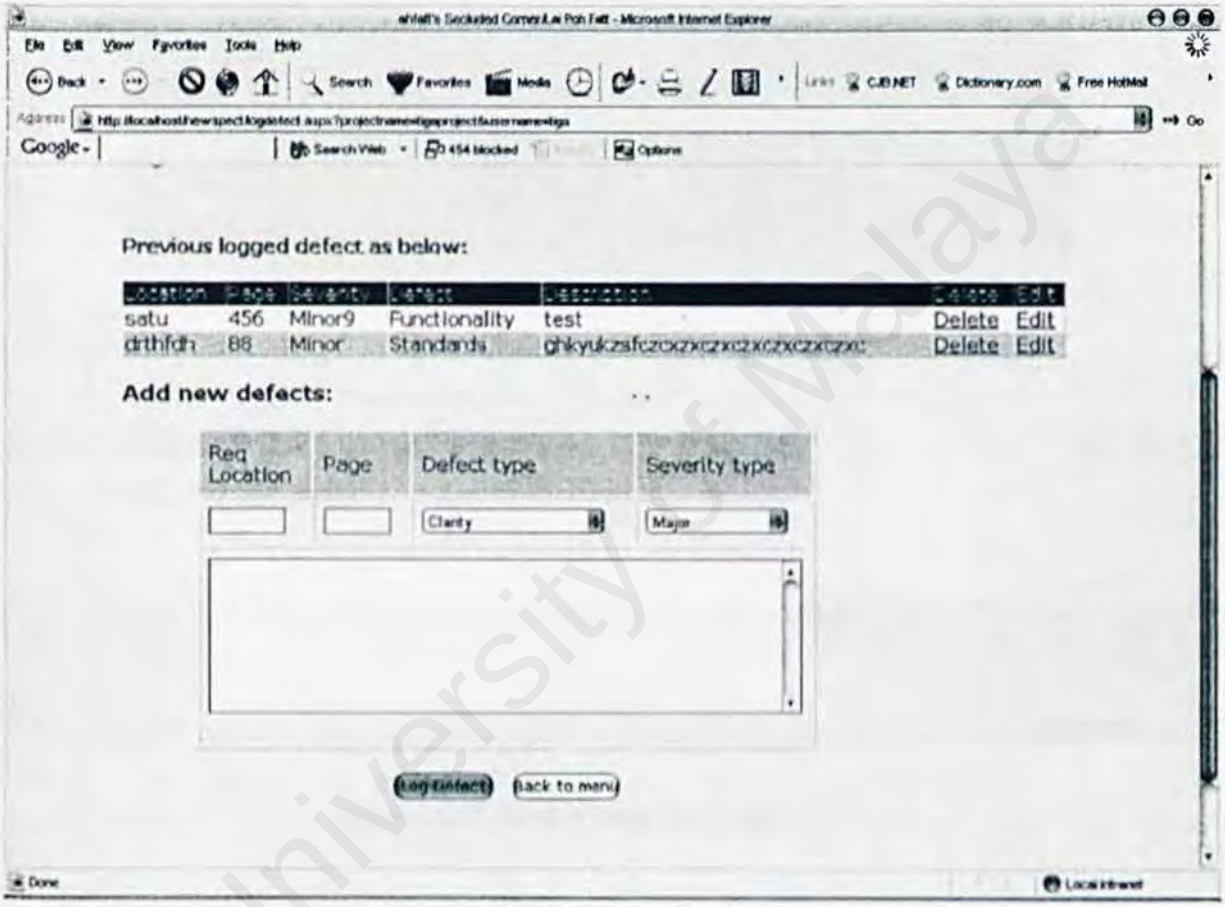


Figure 4-11: Log Defect.

4.12 Review Discovery Logs

This guide will help you to use all the function in **Review Discovery Logs.**

1. Move your mouse to [Review Discovery Logs] and click it. Your information will appear as shown in figure 4-12.

2. Choose the inspector you wish to review from the dropdown list provided and click on [Review Now].
3. System will display a datagrid showing the inspector's work.
4. Click at [Back to menu] will bring back to menu page.

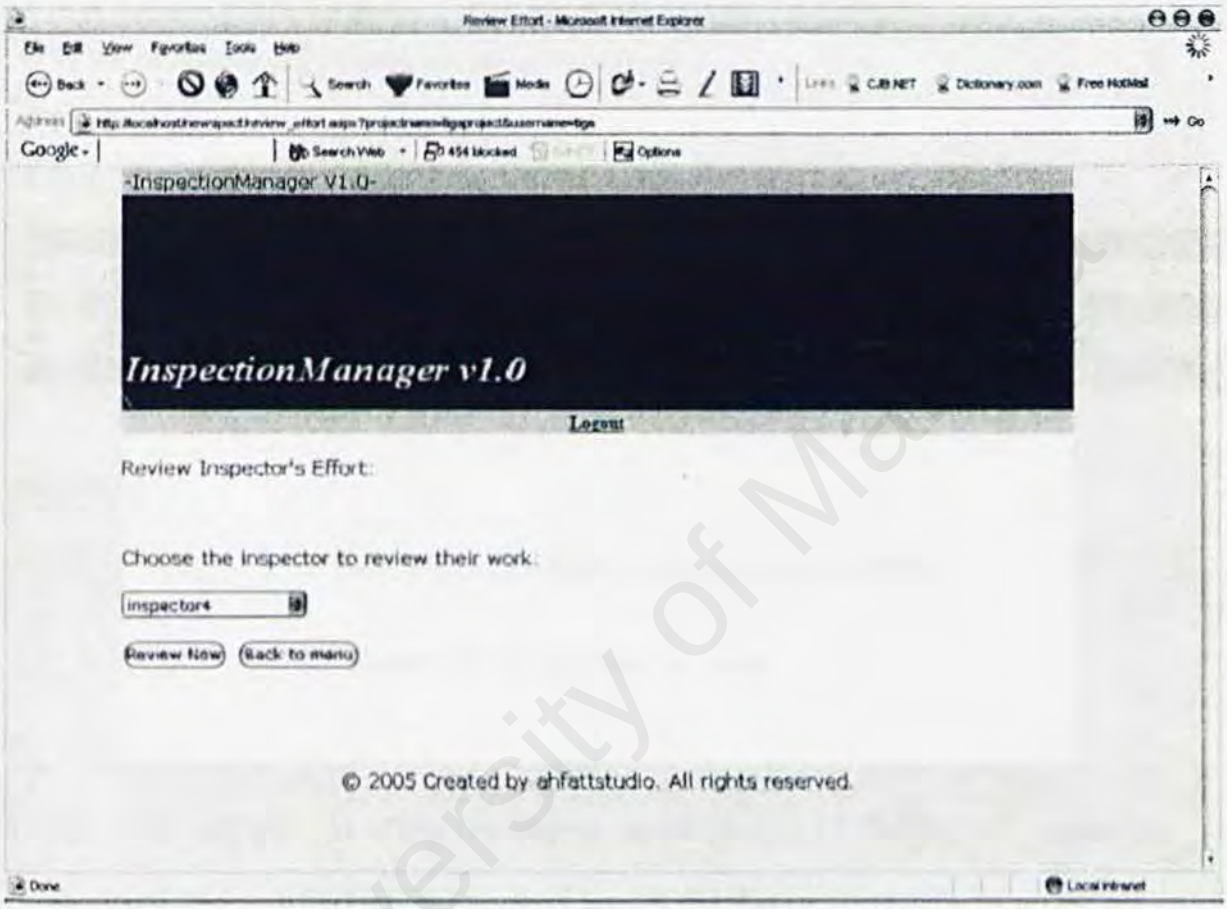


Figure 4-12: Review Discovery Logs.

4.13 Merge Discovery Logs

This guide will help you to use all the function in **Merge Discovery Logs.**

1. Move your mouse to [Review Discovery Logs] and click it. Your information will appear as shown in figure 4-13.

2. Click on [Merge now!] to analyze on the defect from the entire inspector and find the high potential defect.
3. System will display a list of high potential defects.

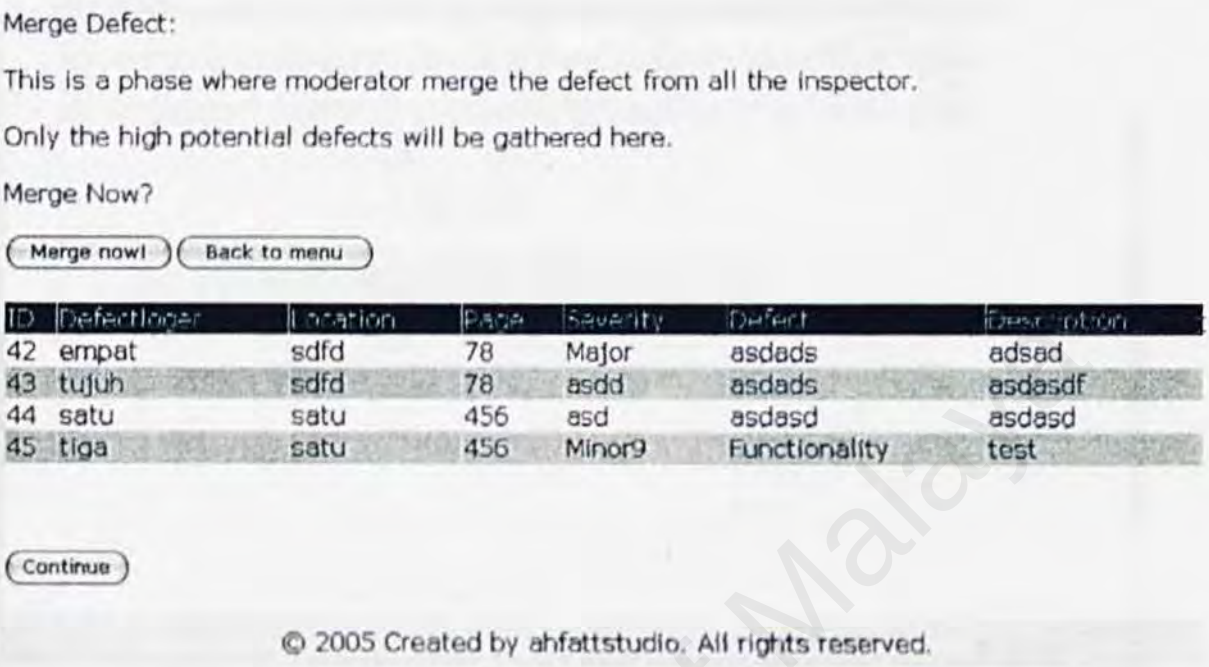


Figure 4-13: Merge Discovery Logs.

4. Click on continue to select the defects to be discuss in the formal inspection meeting by put a tick on the textbox beside the defect.
5. Click at [Select as final defect] and system will show you the defect you've selected in a datagrid.
6. Click at the [Formal Defect] will bring you to the log formal defect page.
7. System will show you the previous logged formal defect (if record exist) and click on the [Log Defect] to save the data bring from the previous page.
8. Click [Back to menu] to back to the menu page.

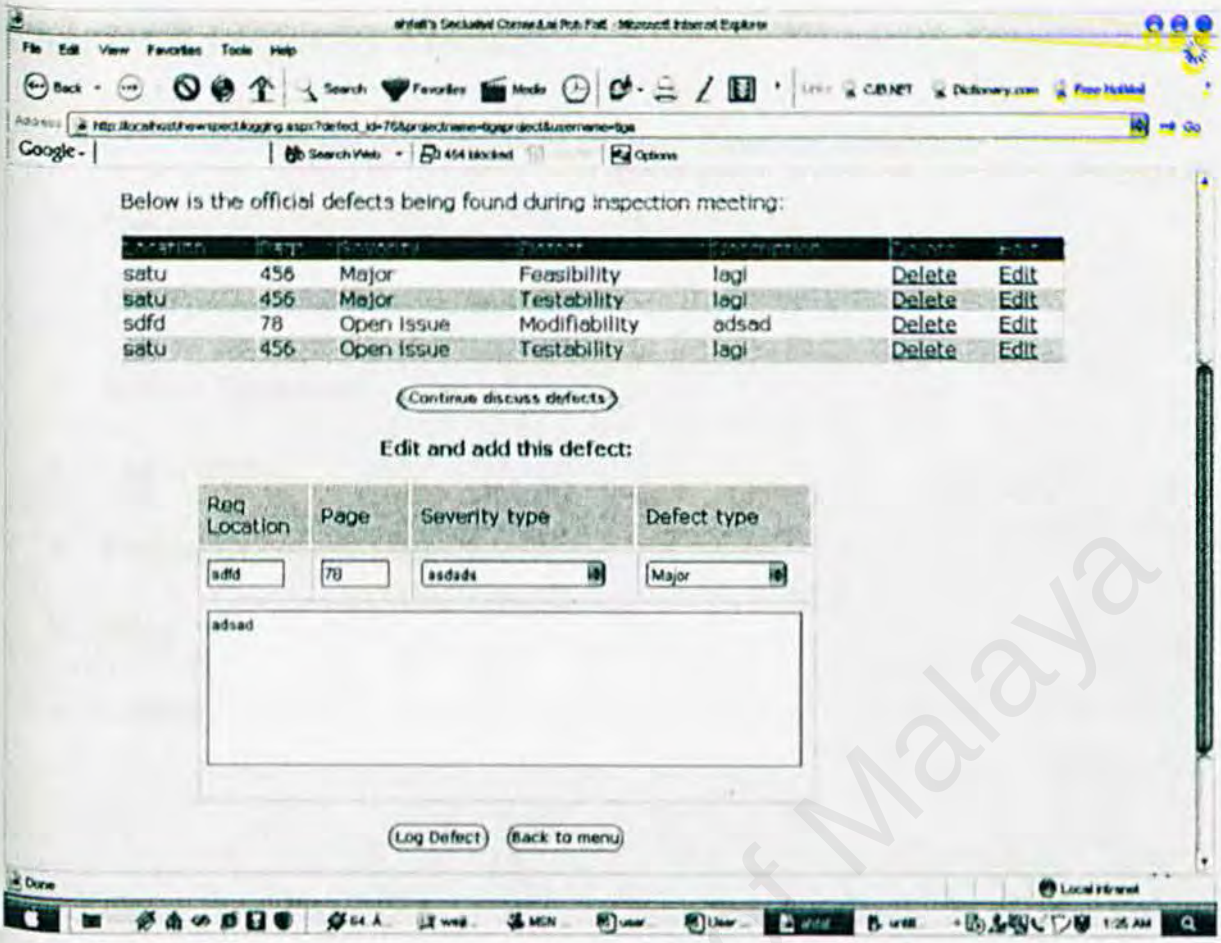


Figure 4-14: Log Formal Defects.

4.14 Print

This guide will help you to use all the function in **Print.**

1. Move your mouse to [Print] and click it.
2. System display four selection for printing: Requirement Document's Checklist, Inspector's Log, Suspected Defect Log and Inspection Issue Log. Click on it to view the details and click [print] to print out the page.

Chapter 5 Inspector Module

This module contents the description of the user function, which are:

- Log in
- View Inspection Information
- Review Document
- Log Defects
- Review Discovery Logs
- Print
- Logout

5.1 Log in

In order to ensure the security and perform the daily task, InspectionManager is required to login as an authorized user. The procedure is similar to the **Moderator Log In** (Refer Section 4.3) and you will be redirect to the **Inspector Main Page.**

5.2 View Inspection Information

This guide will help you view the inspection information, the procedure is similar with the guide at section 4.9

5.3 Review Document

This guide will help you view the requirement document, the procedure is similar with the guide at section 4.10

5.4 Log Defects

This guide will help you to log defects found, the procedure is similar with the guide at section 4.11

5.5 Review Discovery Log

This guide will help you to view your effort; the procedure is similar with the guide at section 4.12

5.6 Print

This guide will help you to view your effort; the procedure is similar with the guide at section 4.14

5.7 Logout

Move your mouse to menu **Log Out** and click it. You will be redirected to the **Main Page.**